Kalle Braun • Jürgen Buchmüller • Frank Seger

Buchserie

Das

Colour-Genie

Buch 1 - 3

Das Colour-Genie Buch

Buch 1

Buch 2

Buch 3

Kalle Braun • Jürgen Buchmüller • Frank Seger

Das Colour-Genie

Buch 1

Das Colour-Genie Buch 1

ISBN 3-88965-003-1

Vorwort

Lieber Colour-Genie Freund!

Das Colour-Genie erfreut sich immer größerer Beliebtheit. So ist es auch nicht verwunderlich, daß immer wieder der Ruf nach einem Buch laut wurde, das speziell für das Colour-Genie geschrieben wurde und das auf die vielfältigen Möglichkeiten des Colour-Genies eingeht.
Nun, das Warten hat ein Ende: Vor Ihnen liegt das "Colour-Genie Buch 1" !
Ein Buch, auf das wir stolz sind, denn es ist vollgepackt mit interessanten Programmen und Informationen und es bietet sowohl dem Anfänger als auch dem fortgeschrittenen Programmierer, sowohl dem Spielefreund als auch dem "ernsthaften" Anwender Programmlistings und Erklärungen, die helfen, das Colour-Genie besser zu verstehen und besser zu nutzen.

Schauen Sie ruhig einmal in das Inhaltsverzeichnis auf den nächsten beiden Seiten, blättern Sie einmal durch das Buch und Sie werden sehen, wie vielfältig der Inhalt ist.

Wir hoffen, daß Sie Ihr Colour-Genie so vielseitig einsetzen, wie es seinen Möglichkeiten entspricht und daß Sie viel Freude mit Ihrem Computer haben werden.
In diesem Sinne wünschen wir Ihnen nun viel Spaß beim Lesen, Abtippen und Ausprobieren!

Kalle Braun
Jürgen Buchmüller
Frank Seger

Bonn, im August 1983

Inhaltsverzeichnis

Zur Gliederung

Wie Sie vielleicht schon aus dem Inhaltsverzeichnis ersehen konnten, haben wir versucht, die einzelnen Teile dieses Buches in eine sinnvolle Reihenfolge zu bringen.

Zu Beginn finden Sie drei interessante Spielprogramme, die alle bereits auf dem 16K-Grundgerät laufen.

Es folgen dann die Listings zweier Spielprogramme, die mit dem "Colour-Compiler" compiliert werden müssen.

Nach den Spielen folgen mehrere Programme und Erläuterungen, die Ihnen ein tieferes Verständnis für Ihren Computer erleichtern sollen, unter anderem einen kompletten Maschinensprache-Monitor in Basic, der es Ihnen ermöglicht, über die Ebenen der Basicprogrammierung auf die Ebene der Maschinenprogrammierung vorzustoßen.

Etliche Kapitel befassen sich mit der Programmierung eines angeschlossenen Druckers - besondere Freude werden Besitzer des STAR-Printers DP 510/515 an den abgedruckten Programmen haben.

Im Anhang finden Sie dann noch eine Liste der Basic-Tokens, eine Dezimal-Hexadezimal Tabelle, sowie eine Liste der verfügbaren TCS-Colour-Genie Software.

**** ACHTUNG ****
In den Listings entspricht das Paragraphenzeichen (§) dem "Klammeraffen" (@), der rechts neben der <P>-Taste liegt, und das große Ä (Ä) dem Aufwärtspfeil ([ ), der zur Exponentialrechnung benötigt wird.

BUMM BUMM

"Bumm Bumm" ist ein einfaches, jedoch sehr lustiges Telespiel. Zwei Spieler spielen gegeneinander. Jeder Spieler hat eine Kanone und muß nun versuchen, seinen Gegner abzuschießen. Dazu braucht man ein gewisses Gefuhl fur Ballistik, denn ein Schuß wird durch die Eingabe des Abschußwinkels und der Abschußgeschwindigkeit gesteuert.
Das Programm soll hier nicht komplett erklärt werden - einige interessante Teile seien jedoch erlautert:

In Zeile 20 wird in das CRTC-Register fur Zeichen-End-Raster eine 0 gePOKEt. Dies bewirkt, daß die FGR-Darstellung nur in halber Hohe erfolgt. Dabei wird der FGR-Speicher zweimal untereinander dargestellt. Diese doppelte Darstellung ist im übrigen leider nicht vermeidbar. In Zeile 690 wird diese CRTC-Programmierung durch POKE 17158,1 wieder ruckgangig gemacht. Siehe hierzu auch Anhang A des Handbuchs "Colour Basic - leicht gelernt".

Erwähnenswert ist auch, wie der Titel "BUMM BUMM" im FGR-Modus gezeichnet wird. Dabei werden nicht alle Zeichen uber einzelne PLOTs erzeugt, sondern alle Zeichen sind in einer 5 mal 8 Matrix codiert. Die Werte dieser Codierung stehen in den DATA-Zeilen 130 bis 160.
Ausgelesen werden die Werte in den Zeilen 330 bis 430.
Die wichtigste Zeile in dieser Routine ist Zeile 380.
Hier wird geprüft, ob das Bit, angegeben durch Variable B, in dem aus der DATA-Zeile gelesenen Wert A gesetzt ist oder nicht. Dazu wird mit den Zweierpotenzen 1 bis 16 verglichen, die in A(0) bis A(4) gespeichert sind (siehe Zeilen 60 bis 100).
Analog sind in den DATA-Zeilen 170 bis 260 die Werte für die Ziffern 0 bis 9 gespeichert, die in der Winkel- u. Geschwindigkeitseingabe benötigt werden.

Für Patrioten sei hier noch erklärt, wie das Deutschlandlied am Ende des Programms erzeugt wird. Nötig hierzu sind die DATA-Zeilen 110 bis 120 und der Programmteil von Zeile 2510 bis 2530 und weiter von 2580 bis 2610. Ein Sprung auf Zeile 2510 spielt das Deutschlandlied ab. Zeile 2540 bis 2570 dienen nur dazu, den Namen des Siegers in allen Farben leuchten zu lassen, darum können diese Zeilen weggelassen werden.

```
10 FCLS
20 POKE17158,0
30 FGR
40 DEFINTA-C,E-V
50 DIMZ(9,4)
60 A=128
70 FORL=7TOOSTEP-1
80 A(L)=A
90 A=A/2
100 NEXTL
110 DATA3,1,3,1,3,1,3,2,3,3,3,3,3,2,3,2,3,4,3,4,3,3,3,3,3,2,2,7,3,1,3,
1,3,6,3,6,3,5,3,5,3,4,3,4,3,3.3,3,3,2,3,2,3,3,3,1,3,5,3,5,3,5,3,5
120 DATA4,1,4,1,4,1,3,7,3,7,3,6,3,5,3,5,3,6,3,6,3,6,3,5,3,5,3,4,3,3,3,
3,3,2,3,2,3,3,3,4,3,5,3,6,3,4,3,2,3,1,3,1,3,3,3,2,3,1,3,1,3,1
130 DATA30,17,17,30,17,17,30,0
140 DATA17,17,17,17,17,17,14,0
150 DATA17,27,21,21,17,17,17,0
160 DATA17,27,21,21,17,17,17,0
170 DATA21,17,17,17,21
180 DATA1,1,1,1,1
190 DATA21,1,21,16,21
200 DATA21,1,5,1,21
210 DATA17,17,21,1,1
220 DATA21,16,21,1,21
230 DATA16,16,21,17,21
240 DATA21,1,1,1,1
250 DATA21,17,21,17,21
260 DATA21,17,21,1,1
270 FCLS
280 FGR
290 RESTORE
300 FORL=1TO63
310 READL1,L2
320 NEXTL
330 FORC=0TO3
340 FORD=0TO7
350 READA
360 FORB=4TOOSTEP-1
370 X=28+C*38-B*5+D
380 IF(AANDA(B))<>A(B)THEN430
390 Y=D*12
400 FCOLOUR4
410 PLOTX,YTOX+5,YTOX+5,Y+12TOX,Y+12TOX,Y
420 PAINTX+1,Y+1,3,4
430 NEXTB,D,C
440 FORL=0TO9
450 FORM=0TO4
460 READZ(L,M)
470 NEXTM,L
480 CLS
490 COLOUR4
500 LGR
```

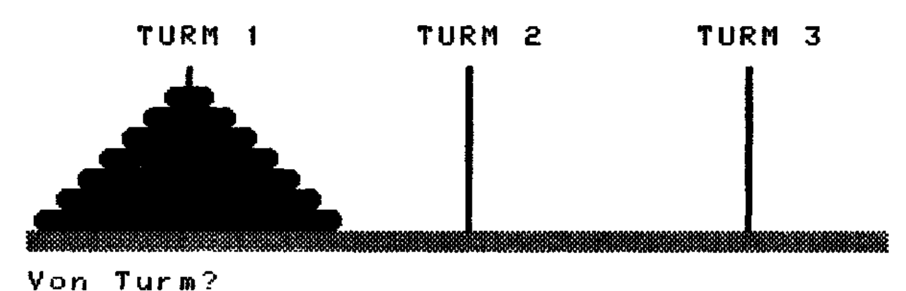
```
510 PRINT"Bei diesem Spiel, dass 2 Spieler gegen-"
520 PRINT"einander spielen, geht es darum, die"
530 PRINT"gegnerische Kanone abzuschiessen."
540 PRINT"Dazu muss jeweils der Abschusswinkel"
550 PRINT"und die Abschussgeschwindigkeit der"
560 PRINT"Kanonenkugel eingegeben werden."
570 PRINT
580 PRINT"Gewonnen hat derjenige, der als erster"
590 PRINT"5 Treffer hat."
600 PRINT
610 PRINT"Reihenfolge der Eingaben:"
620 PRINT"Winkel: 0 bis 90 Grad"
630 PRINT"Geschwindigkeit: 0 bis 99 Meter/sec."
640 PRINT
650 PRINT
660 PRINT
670 PRINT"Geben Sie jetzt bitte Ihre Namen ein:"
680 PRINT
690 POKE17158,1
700 COLOUR5
710 INPUT"NAME DES 1. KANONIERS";C$
720 COLOUR2
730 INPUT"NAME DES 2. KANONIERS";B$
740 IFC$=B$THEN700
750 WF=RND(2)
760 IFS1>4ORS2>4THEN2290
770 GOSUB1710
780 P=18440
790 Z=S1
800 GOSUB2090
810 P=18470
820 Z=S2
830 GOSUB2090
840 IFWF=2THEN1210
850 P=18434
860 GOSUB2180
870 W1=A*10
880 P=P+1
890 GOSUB2180
900 W1=W1+A
910 P=18437
920 GOSUB2180
930 G1=A*10
940 P=P+1
950 GOSUB2180
960 G1=G1+A
970 GOSUB1430
980 XP=P1+11
990 YP=Q1-8
1000 GOSUB1570
```

```
1010 IFV<2THENWF=0:GOSUB2350:GOTO1210
1020 IFXO<80THEN1400
1030 S1=S1+1
1040 WF=1
1050 SOUND7,247
1060 SOUND6,31
1070 SOUND8,16
1080 SOUND9,0
1090 SOUND12,150
1100 SOUND13,9
1110 FORA=1TO100
1120 FCOLOURRND(3)+1
1130 CIRCLEXO,YO,AAND7
1140 SOUND6,AAND15
1150 NEXTA
1160 FCOLOUR1
1170 FORA=0TO8
1180 CIRCLEXO,YO,A
1190 NEXTA
1200 GOTO760
1210 P=18464
1220 GOSUB2180
1230 W1=A*10
1240 P=P+1
1250 GOSUB2180
1260 W1=W1+A
1270 P=18467
1280 GOSUB2180
1290 G1=A*10
1300 P=P+1
1310 GOSUB2180
1320 G1=G1+A
1330 GOSUB1430
1340 DX=-DX
1350 XP=P2-1
1360 YP=Q2-8
1370 GOSUB1570
1380 IFV<2THENWF=0:GOSUB2350:GOTO850
1390 IFXO>80THEN1030
1400 S2=S2+1
1410 WF=2
1420 GOTO1050
1430 W1=W1/57
1440 SOUND7,247
1450 SOUND8,10
1460 SOUND9,10
1470 FORL=31TO0STEP-1
1480 SOUND6,L
1490 FORA=1TO4
1500 NEXT
```

```
1510 NEXT
1520 SOUND7,252
1530 SOUND1,1
1540 DX=G1/10*COS(W1)
1550 DY=-G1/10*SIN(W1)
1560 RETURN
1570 SOUND0,(YPAND255)
1580 SOUND2,255-(YPAND255)
1590 IFXP<0ORYP<0THEN1640
1600 V=CPOINT(XP, YP)
1610 PLOTXP,YP
1620 XO=XP
1630 YO=YP
1640 XP=XP+DX
1650 YP=YP+DY
1660 DY=DY+.1
1670 NPLOTXO,YO
1680 IFXP<0ORYP>95ORXP>159THENRETURN
1690 IFV=0THEN1570
1700 RETURN
1710 FCLS
1720 FGR
1730 FCOLOUR2
1740 P1=RND(40)
1750 P2=RND(40)+110
1760 Y=RND(10)+50
1770 Y=Y*5
1780 Y1=RND(20)-10
1790 FORX=1TO159
1800 PLOTX,Y/5TOX,96
1810 IF(X-P1)<10IF(X-P1)>=0Q1=Y/5:GOTO1840
1820 IF(X-P2)<10IF(X-P2)>=0Q2=Y/5:GOTO1840
1830 IFY+Y1>150IF Y+Y1<480Y=Y+Y1
1840 IF RND(100)<10Y1=RND(20)-10
1850 NEXTX
1860 X=P1
1870 Y=Q1
1880 GOSUB1930
1890 X=P2
1900 Y=Q2
1910 GOSUB2010
1920 RETURN
1930 FCOLOUR3
1940 FORZ=0TO6
1950 PLOTX+Z,Y-ZTOX+6,Y-Z
1960 NEXTZ
1970 FCOLOUR4
1980 PLOTX+7,Y-4TOX+10,Y-7
1990 PLOTX+7,Y-5TOX+10,Y-7
2000 RETURN
```

```
2010 FCOLOUR4
2020 FORZ=0TO6
2030 PLOTX+10-Z,Y-ZTOX+4,Y-Z
2040 NEXTZ
2050 FCOLOUR3
2060 PLOTX+3,Y-4TOX,Y-7
2070 PLOTX+3,Y-5TOX,Y-7
2080 RETURN
2090 A=INT(Z/10)
2100 FORY=0TO4
2110 POKEP+Y*40,Z(A,Y)
2120 NEXTY
2130 A=Z-A*10
2140 FORY=0TO4
2150 POKEP+Y*40+1,Z(A,Y)
2160 NEXTY
2170 RETURN
2180 FORY=0TO160STEP40
2190 POKEP+Y,42
2200 A$=INKEY$
2210 IFA$>="0"ANDA$<="9"THEN2240
2220 NEXTY
2230 GOTO2180
2240 A=VAL(A$)
2250 FORY=0TO4
2260 POKEP+Y*40,Z(A,Y)
2270 NEXTY
2280 RETURN
2290 LGR
2300 CLS
2310 IFS1>4THENS$=C$ELSES$=B$
2320 PRINTS$;" HAT GEWONNEN"
2330 GOSUB2500
2340 RUN
2350 SOUND7,247
2360 SOUND8,15
2370 SOUND9,0
2380 FORL=0TO31
2390 FCOLOURRND(3)+1
2400 CIRCLEXO,YO,LAND3
2410 SOUND6,LAND7
2420 NEXTL
2430 FCOLOUR1
2440 FORL=0TO3
2450 CIRCLEXO,YO,L
2460 NEXTL
2470 SOUND7,255
2480 FCOLOUR3
2490 RETURN
2500 RESTORE
```

```
2510 FORL=1TO63
2520 READO,T
2530 PLAY(1,O,T,14)
2540 F=RND(16)
2550 FORM=&HF000TO&HF000+LEN(S$)
2560 POKEM,F
2570 NEXTM
2580 FORM=1TO100-LEN(S$)
2590 NEXTM
2600 NEXTL
2610 PLAY(1,1,1,0)
2620 RETURN
```



Türme von Hanoi ist ein bekanntes Denkspiel. Die Regeln werden im Programm selbst erklärt.
Eine Besonderheit für Denkfaule: Das Programm löst das Problem auf Wunsch auch selbst - und zwar mit der Minimalanzahl von 127 Zügen. Die Rechnerlösung ist in den Zeilen 20 und 30 enthalten - passen Sie hier besonders auf, daß Sie sich nicht vertippen.
Programmlisting:

```
10 CLS:CLEAR 500:O=0:A=7:CHAR4
20 RD$="12132312313212132321312312132312313212312321313212132312313212
132321312312132321313212312321312312132312313212132321312312132312"
30 Z2$="31321231232131321213231231321231232131231213232131321231232131
3212132312313212132321312312132312313212312321313212132312313212"
40 RD=1
50 PRINT"Die Scheiben des linken Turms muessen"
60 PRINT"in der gleichen Reihenfolge auf der"
70 PRINT"mittleren Scheibe liegen!!"
80 PRINT"Dabei darf niemals eine grosse Scheibe"
90 PRINT"auf einer kleinerer liegen!!"
100 PRINT$720,"1 - Rechner spielt":PRINT$760,"2 - Sie spielen"
110 X1$=INKEY$:IFX1$<>"1"ANDX1$<>"2"THEN110
120 IFX1$="1"THEN160
130 FKEY1="1
140 FKEY2="2
150 FKEY3="3
160 DIMB$(3,8),T(3,8),F(3,7):C1=1:O=0
170 POKE16409,70:POKE16410,100
180 REM
190 REM FARBEN
200 REM
210 FOR J=1 TO 7
220 READ F(1,J)
```

```
230 NEXT
240 REM
250 REM POSITIONEN
260 REM
270 FOR C=1 TO 3
280 FOR K=7 TO 1 STEP -1
290 READ T(C,K)
300 NEXT K
310 NEXT C
320 REM
330 REM SCHEIBEN-STRING'S ERSTELLEN
340 REM
350 FOR C=1 TO  A
360 B$(1,C)=CHR$(238)+STRING$(C1,202)+CHR$(245)
370 C1=C1+2
380 NEXT C
390 CLS
400  GOSUB 640
410 REM
420 REM SCHEIBEN SETZEN
430 REM
440 FOR B=1 TO A
450 COLOUR F(1,B)
460 PRINT$647-A*40+B*40-B,B$(1,B);
470 NEXT:PRINT:PRINT
480 REM
490 REM SPIELEN
500 REM
510 COLOUR  1
520 PRINT$848,"Spielzug Nr."O;
530 IFX1$="1"THENGOSUB1280:GOTO600
540 PRINT$768,STRING$(4,32);
550 PRINT$760,"Von Turm";:INPUTE
560 IFE<1ORE>3THEN550
570 PRINT$785,STRING$(4,32);
580 PRINT$775,"nach Turm";:INPUTF
590 IFF<1ORF>3THEN580
600 O=O+1:GOTO 860
610 REM
620 REM TUERME AUFBAUEN
630 REM
640 COLOUR1:FOR B=1 TO A+1
650 PRINT$687-B*40,CHR$(225);
660  PRINT$700-B*40,CHR$(225);
670 PRINT$713-B*40,CHR$(225);
680 NEXT
690 PRINT$680,STRING$(40,196);
700 PRINT$285,"TURM 1";:PRINT$298,"TURM 2";:PRINT$311,"TURM 3";
710 PRINT$88,STRING$(3," ");" TUERME VON HANOI ";STRING$(3," ");
720 RETURN
730 REM
740 REM FARBEN FUER SCHEIBEN
750 REM
760 DATA 14,7,12,9,6,15,3
770 REM
780 REM DATAS FUER POSITIONEN
790 REM
```

```
800 DATA 647,607,567,527,487,447,407
810 DATA 660,620,580,540,500,460,420
820 DATA 673,633,593,553,513,473,433
830 REM
840 REM SUBROUTINE FUER SCHEIBEN-LOESCHEN
850 REM
860 FOR V=1 TO A
870 IF B$(E,V) > "" THEN 890
880 NEXT V
890 FOR V1=1 TO A
900 IF B$(F,V1) > "" THEN 910 ELSE NEXT V1
910 V1=V1-1
920 IF V=8 THEN 1190
930 IF V1<7 AND B$(E,V) < B$(F,V1+1) THEN 1140
940 F(F,V1)=F(E,V)
950 FOR C=1 TO INT(LEN(B$(E,V)))/2+1
960 PRINT$T(E,V)-INT(LEN(B$(E,V))/2)+C-2,CHR$(32);
970 PRINT$T(E,V)-INT(LEN(B$(E,V))/2)+LEN(B$(E,V))-C+1,CHR$(32);
980 NEXT:COLOUR1
990 PRINT$T(E,V),CHR$(225);
1000 REM
1010 REM SUBROUTINE FUER SCHEIBEN-SETZEN
1020 REM
1030 COLOUR F(F,V1)
1040 PRINT$T(F,V1)-INT(LEN(B$(E,V))/2),CHR$(238);
1050 PRINT$T(F,V1)+INT(LEN(B$(E,V))/2),CHR$(245);
1060 FOR C=1 TO INT(LEN(B$(E,V))/2)
1070 PRINT$T(F,V1)-INT(LEN(B$(E,V))/2)+C,CHR$(202);
1080 PRINT$T(F,V1)-INT(LEN(B$(E,V))/2)        +LEN(B$(E,V))-C-1,CHR$(2
02);
1090 NEXT
1100  IF F=2 AND V1=1 THEN 1250
1110 B$(F,V1)=B$(E,V):B$(E,V)=""
1120 REMT1(E)=T1(E)+1:T1(F)=T1(F)-1
1130 GOTO 510
1140 FOR Q=1 TO 5
1150 COLOUR 4:PRINT$0,"Keine grosse Scheibe auf eine kleinere !";
1160 FOR Q1=1 TO 70:NEXT Q1
1170 PRINT$0,STRING$(40,128);
1180 NEXT Q:GOTO 510
1190 FOR  Q=1 TO 6
1200 COLOUR 4:PRINT$0,"Turm"E"ist leer !";
1210 FOR Q1=1 TO 70:NEXT Q1
1220 PRINT$0,STRING$(17,128);
1230 NEXT Q:GOTO 510
1240 FKEY1="LIST":FKEY2="RUN":FKEY3="AUTO"
1250 IF  O > 140 THEN 1270
1260 CLS:PRINT$440,"BRAVO-Mit"O"Zuegen geschafft !":PRINT:PRINT:END
1270 CLS:PRINT$440,"Mit"O"Zuegen geschafft-da fehlt die     Uebung-wei
termachen !":PRINT:PRINT:END
1280 IFRD>128THENRD$=Z2$:RD=1
1290 E=VAL(MID$(RD$,RD,1)):F=VAL(MID$(RD$,RD+1,1)):RD=RD+2:RETURN
```

Schiffe versenken

Das im folgenden gelistete Programm ist eine Abwandlung des hinreichend bekannten Spiels "Schiffe versenken". Dieses Programm hat zwar nicht den hochsten Schwierigkeitsgrad, aber die Ausgabe des Feldes wird mit einigen grafischen Tricks gemacht. Am auffalligsten ist die Tatsache, das die Wellen tatsachlich rollen. Hier einige Worte dazu.

In den Zeilen 1210 bis 1260 wird in dem Bereich von 409FH bis 40E0H ein kleines Maschinenspracheprogramm initialisiert, das bei jedem 16. Aufruf den gesamten Bildschirm nach Zeichen zwischen 128 und 159 absucht und diese um 1 erhoht. Durch CALL 43D3 wird diese Routine in den Tastatur-DCB eingehangt (siehe Kapitel uber die DCBs). Bei jeder 16. Tastaturabfrage werden nun die Wellen bewegt. Mit CALL 43DA wird dies wieder abgeschaltet.

Noch einige Worte zum Spielablauf:

Zunachst stellt der Rechner seine Schiffe auf. Diese sind:
1 Schlachtschiff (4 Kastchen lang)
2 Zerstorer (3 Kastchen lang)
3 Kreuzer (2 Kastchen lang) und
4 Aufklarer (1 Kastchen lang)

Sie konnen nun das rote Fadenkreuz, das oben links auf dem Wasser erscheint, mit den 4 Pfeiltasten bewegen. Wenn Sie eine Position erreicht haben, auf die Sie schießen wollen, drucken Sie <RETURN>. Der Rechner teilt Ihnen dann unten rechts mit ob Sie getroffen oder nur ins Wasser geschossen haben. Außerdem hort man bei einem Treffer eine Explosion und der zerschossene Teil eines Schiffes erscheint. Alle Schiffe liegen waagerecht, mit dem Bug nach rechts. Außerdem sind Sie an Ihrer Farbe erkennbar. In der untersten Bildschirmzeile sehen Sie standig eine Auswertung Ihrer bisherigen "Arbeit": die Anzahl der abgegebenen Schusse und die Anzahl der Treffer werden dort angezeigt.

Wenn Sie alle Schiffe zerstort haben, erscheint noch eine Auswertung, wie gut (oder wie schlecht) Sie gespielt haben.

Programmlisting:

```
10 CLS
20 DEFINTA-Z
30 DIMF1(10,10),F(10,10),R(4)
40 GOSUB700
50 GOTO1280
60 DATA0,8640,30,4620
70 DATA16416,8608,1054,4618
80 DATA0,18480,135,-31613
90 DATA4104,18472,391,-31742
100 DATA0,4620,16609,8608
110 DATA1026,4618,225,8640
120 DATA128,-31613,4216,18472
130 DATA256,-31742,120,18480
140 DATA16416,8608,1054,4618
150 DATA0,8640,30,4620
160 DATA4104,18472,391,-31742
170 DATA0,18480,135,-31613
180 DATA1026,4618,225,8640
190 DATA0,4620,16609,8608
200 DATA256,-31742,120,18480
210 DATA128,-31613,4216,18472
220 DATA225,16416,8865,28
230 DATA225,0,17280,60
240 DATA120,4104,-14296,7
250 DATA120,0,-28576,15
260 DATA30,1026,12810,193
270 DATA30,0,9240,195
280 DATA135,256,3074,240
290 DATA135,128,-30330,112
300 DATA225,0,17280,60
310 DATA225,16416,8865,28
320 DATA120,0,-28576,15
330 DATA120,4104,-14296,7
340 DATA30,0,9240,195
350 DATA30,1026,12810,193
360 DATA135,128,-30330,112
370 DATA135,256,3074,240
380 DATA1024,1540,1031,1285
390 DATA0,0,4336,-32544
400 DATA0,2056,2056,4127
410 DATA0,0,0,16512
420 DATA256,7175,13073,25894
430 DATA-16384,6256,-6524,-11598
440 DATA1024,1540,1286,1285
450 DATA0,0,-32768,-32672
460 DATA0,0,0,4127
470 DATA0,0,0,16512
480 DATA0,0,0,0
490 DATA0,0,0,0
500 DATA0,0,0,0
510 DATA0,0,0,0
520 DATA0,0,0,0
530 DATA0,0,0,0
540 DATA-251,-1,-103,-1
```

```
550 DATA-128,-1,-103,-1
560 DATA-244,-1,-104,-1
570 DATA-224,-12056,16544,128
580 DATA25931,13094,7184,7
590 DATA-11566,-6606,6156,240
600 DATA-251,-15393,-14463,-1
610 DATA-128,-30745,-28799,-8273
620 DATA-4852,25065,-4208,-13
630 DATA-16608,-28504,16544,128
640 DATA0,0,0,0
650 DATA0,0,0,0
660 DATA6407,16673,-32447,-127
670 DATA12480,1032,516,-254
680 DATA-32383,16705,6433,7
690 DATA514,1028,12296,192
700 PRINT"Ich denke mir eine Aufstellung aus!"
710 FORX=0TO511STEP2
720 READA
730 POKEX-3072,AAND255
740 POKEX-3071,INT(A/256)AND255
750 NEXTX
760 GOSUB1210
770 FORX=1TO10
780 FORY=1TO10
790 F(X,Y)=0
800 F1(X,Y)=0
810 NEXTY,X
820 FORZ=1TO10
830 READL
840 FORX=1TO4
850 R(X)=0
860 NEXTX
870 SX=RND(10)
880 SY=RND(10)
890 FORX=1TO2
900 IFR(X)<>0THENNEXTX:GOTO840
910 A=RND(2)
920 IFR(A)<>0THEN910
930 R(A)=1
940 TX=SX
950 TY=SY
960 FORX=1TOL
970 IFF(TX,TY)<>0THEN890
980 IFTX-1>0THENIFF(TX-1,TY)<>0THEN890
990 IFTX+1<11THENIFF(TX+1,TY)<>0THEN890
1000 IFTY-1>0THENIFF(TX,TY-1)<>0THEN890
1010 IFTY+1<11THENIFF(TX,TY+1)<>0THEN890
1020 ONAGOSUB1130,1150,1170,1190
1030 IFTX<1ORTX>10ORTY<1ORTY>10THEN890
1040 NEXTX
1050 TX=SX
1060 TY=SY
1070 FORX=1TOL
1080 F(TX,TY)=L
```

```
1090 ONRGOSUB1130,1150,1170,1190
1100 NEXTX
1110 NEXTZ
1120 RETURN
1130 TX=TX+1
1140 RETURN
1150 TX=TX-1
1160 RETURN
1170 TY=TY+1
1180 RETURN
1190 TY=TY-1
1200 RETURN
1210 FORX=1TO66
1220 READA
1230 POKE&H439E+X,A
1240 NEXT
1250 RETURN
1260 DATA58,158,67,60,230,15,50,158,67,194,227,3,229,197,33,0,68,126,2
54,128,56,19,254,160,48,15,60,60,71,230,15,254,2,62,0,48,2,62,240,128,
119,35,124,254,72,32,226,193,225,195,227,3,33,159,67,34,22,64,201,33,2
27,3,34,22,64,201
1270 DATA4,3,3,2,2,2,1,1,1,1
1280 COLOUR1
1290 CLS
1300 PRINT§7,"**  SCHIFFE VERSENKEN  **"
1310 TR=0
1320 SH=0
1330 H1$=CHR$(160)+CHR$(161)
1340 H2$=CHR$(176)+CHR$(177)
1350 B1$=CHR$(162)+CHR$(163)
1360 B2$=CHR$(178)+CHR$(179)
1370 PRINT§40,STRING$(39,217);
1380 PRINT§64,CHR$(242);
1390 FORX=2TO23
1400 PRINT§X*40+24,CHR$(225);
1410 NEXT
1420 COLOUR3
1430 PRINT§146,"Schlachtschiff";
1440 PRINT§186,H1$;H1$;H1$;B1$;
1450 PRINT§226,H2$;H2$;H2$;B2$
1460 COLOUR1
1470 PRINT§306,"Zerstoerer";
1480 PRINT§346,H1$;H1$;B1$;
1490 PRINT§386,H2$;H2$;B2$;
1500 COLOUR4
1510 PRINT§466,"Kreuzer";
1520 PRINT§506,H1$;B1$;
1530 PRINT§546,H2$;B2$;
1540 COLOUR2
1550 PRINT§626,"Aufklaerer";
1560 PRINT§666,B1$;
1570 PRINT§706,B2$;
1580 COLOUR7
1590 PRINT§786,"Wasser";
1600 CALL43DA
1610 PRINT§826,;
1620 FORX=1TO7
```

```
1630 PRINTCHR$(128);CHR$(129);
1640 NEXTX
1650 PRINT$866,;
1660 FORX=1TO7
1670 PRINTCHR$(144);CHR$(145);
1680 NEXTX
1690 FORX=1TO10
1700 FORY=1TO10
1710 AD=&H4402+Y*80+2*X-2
1720 POKEAD-21504,1
1730 POKEAD-21503,1
1740 POKEAD-21464,1
1750 POKEAD-21463,1
1760 POKEAD,128
1770 POKEAD+1,129
1780 POKEAD+40,144
1790 POKEAD+41,145
1800 NEXTY,X
1810 CALL43D3
1820 X=1
1830 Y=1
1840 PRINT$920,SH;"Schuesse,";TR;"Treffer";
1850 CU=&H4400+Y*80+2*X
1860 POKECU-21504,2
1870 POKECU-21503,2
1880 POKECU-21464,2
1890 POKECU-21463,2
1900 POKECU,188
1910 POKECU+1,189
1920 POKECU+40,190
1930 POKECU+41,191
1940 KB$=INKEY$
1950 IFKB$=""THEN1940
1960 IFF1(X,Y)<>0THENBF=0:GOSUB2140:GOTO2080
1970 CALL43DA
1980 CH=PEEK(&H4400+826)
1990 POKECU-21504,1
2000 POKECU-21503,1
2010 POKECU-21464,1
2020 POKECU-21463,1
2030 POKECU,CH
2040 POKECU+1,CH+1
2050 POKECU+40,CH+16
2060 POKECU+41,CH+17
2070 CALL43D3
2080 IFKB$=CHR$(13)THEN2480
2090 IFKB$="Ä"THENIFY>1THENY=Y-1
2100 IFKB$=CHR$(10)THENIFY<10THENY=Y+1
2110 IFKB$=CHR$(8)THENIFX>1THENX=X-1
2120 IFKB$=CHR$(9)THENIFX<10THENX=X+1
2130 GOTO1840
2140 IFX=10THENCH=168:GOTO2170
2150 IFF(X+1,Y)=0THENCH=168:GOTO2170
2160 CH=166
```

```
2170 AD=&H4400+Y*80+2*X
2180 IFBF=0THEN2360
2190 SOUND6,RND(10)+21
2200 SOUND7,7
2210 SOUND8,16
2220 SOUND12,200
2230 SOUND13,9
2240 FORT=1TO10
2250 POKEAD-21504,6
2260 POKEAD-21503,6
2270 POKEAD-21464,6
2280 POKEAD-21463,6
2290 POKEAD,164
2300 POKEAD+1,165
2310 POKEAD+40,180
2320 POKEAD+41,181
2330 GOSUB2360
2340 NEXTT
2350 RETURN
2360 CO=6-F1(X,Y)
2370 POKEAD-21504,CO
2380 POKEAD-21503,CO
2390 POKEAD-21464,CO
2400 POKEAD-21463,CO
2410 POKEAD,CH
2420 POKEAD+1,CH+1
2430 POKEAD+40,CH+16
2440 POKEAD+41,CH+17
2450 RETURN
2460 IFKBAND64=64THENIFX<10THENX=X+1
2470 GOTO1840
2480 IFF1(X,Y)<>0THEN1840
2490 F1(X,Y)=F(X,Y)
2500 IFF1(X,Y)=0THENPRINT$946,"Wasser    ";ELSEPRINT$946,"Getroffen";:B
F=1:GOSUB2140:TR=TR+1
2510 SH=SH+1
2520 IFTR<20THEN1840
2530 CALL43DA
2540 CLS
2550 COLOUR1
2560 PRINT"Sie haben mit";SH;"Schuessen alle "
2570 PRINT"Schiffe getroffen!!"
2580 IFSH=20THENPRINT"Super, Sie sind ein hervorragender":PRINT"Admira
l"
2590 IFSH<30THENPRINT"Sie haben gute Chancen, eine See-":PRINT"schlach
t zu gewinnen"
2600 IFSH<50THENPRINT"Sie sollten noch etwas ueben!"
2610 IFSH>49THENPRINT"Auweia, da haben Sie aber viel Pech":PRINT"gehab
t!!!!"
2620 PRINT$960,"Neues Spiel??? (J/N)";
2630 NS$=INKEY$
2640 IFNS$<>"J"ANDNS$<>"N"THEN2630
2650 IFNS$="J"THENRUN
2660 PRINT$400,"Schade"
2670 END
```

HEKTIK - ein komplexes Spielprogramm analysiert

"Hektik" wird schon seit langerer Zeit fur 39.- DM verkauft und gehort zu den beliebtesten TCS-Spielprogrammen.
Vielleicht haben Sie es schon, dann werden Sie beim Spielen sicher eine Menge Spass gehabt haben. Wahrscheinlich haben Sie sich auch gefragt, wie man ein derartiges Spiel programmiert.
Diese Frage soll nicht unbeantwortet bleiben!
Zwei Voraussetzungen mußen allerdings erfullt sein:
Ihr Colour-Genie muß 32K RAM haben, und Sie mussen den "Colour-Compiler" besitzen. Denken Sie bitte nicht, daß wir Sie auf diese Weise zum Kauf des Compilers drangen mochten - der Compiler ist fur schnelle Videospiel-Programmierung einfach unerlasslich, es sei denn Sie programmieren in Maschinensprache.
Es folgt nun das Basic-Listing (also der "Source-Code") von Hektik. Wichtig ist, daß Sie beim Einschalten die MOD SEL-Taste gedruckt halten, da sonst der verfugbare Speicher nicht ausreicht !
Nach dem Listing finden Sie eine detaillierte Erklarung des Programms. Bezuglich der Bedienung des "Colour-Compilers" informieren Sie sich ggf. in dessen Handbuch.
Das Basicprogramm geben Sie bitte genauso ein, wie Sie es hier aufgelistet finden. Die Grafikzeichen, die im Listing vorkommen, geben Sie, wie in Kap. 6 des Handbuchs "Colour Basic - leicht gelernt" beschrieben, mit der MOD SEL-Taste ein.*

Programmlisting:

```
10 DATA-127,-32383,-127,-32383
20 DATA-1,-6400,231,-1
30 DATA-61,-6400,231,-1
40 DATA-15487,-6400,231,-1
50 DATA-15487,-15616,231,-1
60 DATA-32383,-32512,129,-61
70 DATA-32383,-32512,129,-32383
80 DATA0,0,0,0
90 DATA-32383,-32257,-32383,-32257
100 DATA-6169,-256,255,-6169
110 DATA-15487,-6400,255,-6169
120 DATA-32383,-15616,231,-6169
130 DATA-32383,-32512,195,-6205
140 DATA-32383,-32512,129,-15487
150 DATA-32383,-32512,129,-32383
160 DATA0,0,0,0
```

```
170 DATA17025.-25,-6247,9342
180 DATA17025.-6334.-26113,15462
190 DATA258,511,1,0
200 DATA0.12736,525.2052
210 DATA-32704.-32513,128,0
220 DATA0.-29693.16560,4128
230 DATA-1538.6264,-31492,1924
240 DATA-1538.6264.18472.3624
250 DATA15900,13106,-25538,-26472
260 DATA31800,-13236.14716,6425
270 DATA-26498,6296,10020,-16352
280 DATA7168.14654.7230,6425
290 DATA6526,6169.-7132,772
300 DATA14336,-25476.14460.-26472
310 DATA-24705.6174,8511,-8159
320 DATA-24705,6174,4628,12308
330 FORL=0TO255 POKEL-3072,PEEK(L+18436)
 NEXTL
340 CHAR2
350 DATA 769,5,769,5,1282,5,1282,5,769,4
,769.4.514,512,770,521,514.0
360 FORL=0TO31 POKE30720+L,PEEK(L+18692)
 NEXTL
370 POKE16409,7 POKE16410,6
380 CLS
390 E=2 M2=3 P1=1
400 GOSUB2070
410 B2=0 P2=0 Q=0 Q1=0
420 GOSUB2580 COLOUR11 FORY=80TO880STEP1
60 FORX=0TO38STEP2
430 POKE17408+X+Y,129 POKE17409+X+Y,137
440 POKEX+Y-4096,6 POKEX+Y-4095,6
450 NEXTX,Y
460 FORX=0TO39 POKE18288+X,202 NEXTX
470 FORY=40TO680STEP160
480 A=Y/20+2 B=37-Y/20
490 FORY1=0TO160STEP40
500 POKE17408+Y+Y1+B,128 POKEY+Y1+B-4096
,3
510 POKE17408+Y+Y1+A,128 POKEY+Y1+A-4096
,3
520 NEXTY1,Y
530 PRINT@920,"PUNKTE            BONUS
    JAEGER ",
540 PRINT@956,M2,
550 IFT=0LETP2=P2+B2
560 SOUND7,56 SOUND8,16 SOUND9,0 SOUND10
,0 SOUND1,2 SOUND3,4 SOUND5,2 SOUND12,10
570 T=0 T2=30719 O2=3 B2=R1*500
580 FORL=0TOE
590 POKE31000+L,L+15 POKE31010+L,1
600 POKE31020+L,1
610 POKE31030+L,3
620 POKE31040+L,32
630 NEXTL
```

```
640 X=19:Y=20:H=32:H1=32:H2=32:X2=1:Y2=0
650 A=PEEK(-1984):IFA=6GOTO380
660 X1=0:Y1=0
670 IF(AAND32)=32LETX1=-1:GOTO710
680 IF(AAND64)=64LETX1=+1:GOTO710
690 IF(AAND08)=08LETY1=-1:GOTO710
700 IF(AAND16)=16LETY1=+1
710 P=17408+X+Y*40
720 IFI1<1GOTO750
730 IFH1=238LETH1=32ELSEIFH1=245LETH1=32
740 IFH2=238LETH2=32ELSEIFH2=245LETH2=32
750 C=C+1:C1=(C/2)AND1
760 IFH>143IFH<146GOTO1120
770 IFH1>143IFH1<146GOTO1120
780 IFB2>0LETB2=B2-1
790 IF(AAND127)<>0POKEP,H:POKEP+40,H1:PO
KEP+40+X2,H2:POKEP-21504,15:POKEP-21464,1
5
800 IFX+X1=0LETX1=0ELSEIFX+X1=39LETX1=0
810 IFX1<>0LETX2=X1:Y2=0
820 IFY1<>0LETY2=Y1:X2=0
830 P1=P+Y1*40:P=P+X1
840 Z=PEEK(P+80):IFZ<>128IFZ<>129IFZ<>13
7IFZ<>135IFZ<>143IFZ<>202LETX1=0
850 IFZ=135GOTO1990ELSEIFZ=143GOTO1990
860 IFPEEK(P1+40)<>128LETY1=0
870 X=X+X1:Y=Y+Y1:P=17408+X+Y*40:IF(AAND
127)<>0LETH=PEEK(P):H1=PEEK(P+40):H2=PEEK
(P+40+X2)
880 IFI1<1GOTO900
890 IFH2=238GOSUB1880ELSEIFH2=245GOSUB18
80ELSEIFH2=230GOSUB1880
900 POKEP-21504,7:POKEP-21464,8
910 IFX2<>0IFH2=32POKEP-21464+X2,0
920 IFX2=-1POKEP,153:POKEP+40,158+C1:POK
EP+39,148
930 IFX2=+1POKEP,152:POKEP+40,150+C1:POK
EP+41,146
940 IFY2<>0POKEP,155+C1*2:POKEP+40,154+C
1*2
950 IF(AAND128)<>128GOTO1070
960 IFX2=0GOTO1050
970 SOUND8,16:SOUND13,9:SOUND0,0:SOUND1,
4-C1
980 POKEP+40+X2,148-(X2+1)+C1
990 IFC1=0GOTO1070
1000 P=P+80+X2:Z=PEEK(P)
1010 IFPEEK(-1920)=1GOTO1050
1020 IFZ>128IFZ<135POKEP,Z+1:GOTO1070
1030 IFZ>136IFZ<143POKEP,Z+1:GOTO1070
1040 GOTO1070
```

```
1050 IFZ>129IFZ<136POKEP,Z-1:GOTO1070
1060 IFZ>137IFZ<144POKEP,Z-1:GOTO1070
1070 Q1=Q1+1:IFQ1>(6-R1)/2LETQ1=0:GOSUB1
400ELSEFORM=1TO300:NEXTM
1080 IFZ2=1LETZ2=0:GOTO1120
1090 COLOURRND(16):PRINT@14,"** HEKTIK *
*";:PRINT@927,P2;"0";:PRINT@942,B2;"0 ";
1100 IFT-1=ELETT=0:R1=R1+1:IFE<5LETE=E+1
:GOTO420ELSE420
1110 GOTO650
1120 SOUND7,56:SOUND6,0:FORL=8TO10:SOUND
L,16:NEXTL:SOUND12,90:SOUND13,9
1130 FORM=1TO3100STEP5:SOUND0,M:SOUND2,M
/2:SOUND4,M/4:NEXTM
1140 SOUND7,255
1150 M2=M2-1:IFM2=0GOTO1180
1160 B2=0
1170 GOTO420
1180 IFR2>P2GOTO1200
1190 R2=P2
1200 FORL=0TO8
1210 IFPEEK(31108+L*10)+PEEK(31109+L*10)
*256<P2GOTO1240
1220 NEXTL
1230 GOTO380
1240 SOUND7,7:FORM=8TO10:SOUNDM,16:NEXTM
1250 SOUND6,0:SOUND12,100:SOUND13,9
1260 CLS
1270 FORM=0TO31:COLOURRND(16):PRINT@210,
"N E U E R  R E K O R D";
1280 PRINT@250,"________________________";
1290 SOUND6,M
1300 NEXTM
1310 FORM=31190TO(31100+L*10)STEP-10:FOR
N=0TO9
1320 POKEM+N,PEEK(M+N-10)
1330 NEXTN:NEXTM
1340 PRINT@568,"Geben Sie Ihren Namen ei
n:"
1350 PRINT@697,"........";:PRINT@697,CHR
$(14);
1360 P=31100+L*10:GOSUB2510
1370 POKE31108+L*10,P2AND255
1380 POKE31109+L*10,P2/256
1390 GOTO380
1400 FORL=31000TO31000+E
1410 V=PEEK(L):W=PEEK(L+10):Q=PEEK(L+40)
1420 IFV=255GOTO1660
1430 IFQ>145IFQ<160LETZ2=1:RETURN
1440 IFI1<1IFQ=238LETQ=32ELSEIFQ=245LETQ
=32ELSEIFQ=230LETQ=32
1450 V1=PEEK(L+20)-2:W1=PEEK(L+30)-2
1460 POKE17408+V+W*40,Q
1470 POKEV+W*40-4096,15
1480 IFQ>128IFQ<144POKEV+W*40-4096,6
```

```
1490 IFV+V1=0POKEL+20,3ELSEIFV+V1=39POKE
L+20,1
1500 A=PEEK(17448+V+W*40):B=PEEK(17368+V
+W*40)
1510 IFW=1POKEL+30,3ELSEIFW=21POKEL+30,1
1520 IFA=135LETQ=129:GOTO1740ELSEIFA=143
LETQ=137:GOTO1740
1530 IFW1=1IFA=128LETV1=0:GOTO1550ELSEIF
A<>202IFA>145LETV1=0:GOTO1550
1540 IFW1=1LETW1=0:IFQ=128POKEL+20,RND(2
)OR1
1550 IFW1=-1IFB=128LETV1=0:GOTO1570ELSEI
FB>145LETV1=0:GOTO1570
1560 IFW1=-1LETW1=0:IFQ=128POKEL+20,RND(
2)OR1
1570 V=V+V1:W=W+W1
1580 POKEL+40,PEEK(17408+V+W*40)
1590 POKE17408+V+W*40,143+RND(2)
1600 POKEV+W*40-4096,5
1610 POKEL,V:POKEL+10,W
1620 IFPEEK(L+40)<144GOTO1660ELSEIFPEEK(
L+40)>145GOTO1660
1630 FORM=31000TO31000+E
1640 IFL<>MIFPEEK(M)=VIFPEEK(M+10)=WPOKE
L+40,PEEK(M+40)
1650 NEXTM
1660 NEXTL
1670 T2=T2+1:IFT2=30752LETT2=30720:IFO2<
6LETO2=O2+1ELSEO2=3
1680 A=PEEK(T2):PLAY(3,O2,A,10):IFA<>0SO
UND8,0:SOUND10,16:SOUND13,9
1690 IFI=1GOTO1840
1700 IFRND(1000)<R1GOTO1720
1710 RETURN
1720 I1=50:I=1
1730 RETURN
1740 IFRND(100)<(R1*10)GOTO1960
1750 SOUND9,15:SOUND3,2
1760 P=V+W*40+17448:Q1=6:FORM=PTO18288ST
EP40
1770 POKEM,Q:POKEM-21504,Q1:Q=PEEK(M+40)
:Q1=PEEK(M-21464):POKEM+40,144:POKEM-2146
4,5
1780 SOUND2,(M/3)AND255
1790 FORN=1TO300:NEXTN
1800 NEXTM:POKEM,Q:POKEM-21504,Q1
1810 POKEL,255:T=T+1
1820 SOUND9,0
1830 P2=P2+E*5:GOTO1660
1840 IFI1>0LETI1=I1-1
1850 IFI1=0 PRINT@699,"   ";:I=0:RETURN
1860 IF(I1AND1)=1 PRINT@699," ● "; ELSE P
RINT@699,"●●";
1870 RETURN
1880 I=0:PRINT@698,"1000";
1890 B$=CHR$(1)+CHR$(3)+CHR$(5)+CHR$(4)+
CHR$(6)+CHR$(5)+CHR$(5)+CHR$(1)+CHR$(1)
1900 FORL=0TO8:C=PEEK(30752+L)
1910 PLAY(3,4,C,14)
1920 FORM=1TO900:NEXTM,L
```

```
1930 PRINT@698,"      ";
1940 H=32:H1=32:H2=32:P2=P2+100:I=0:I1=0
:A=8
1950 RETURN
1960 POKE17448+V+W*40,129+(VAND1)*8
1970 FORM=7TO1STEP-1:PLAY(2,4,M,15):FORN
=1TO300:NEXTN:NEXTM:SOUND9,0
1980 GOTO1530
1990 IFH2=128IFPEEK(P-40)<>128LETH2=32
2000 POKEP-X1,H:POKEP+40-X1,H1:POKEP+40-
X1+X2,H2
2010 X=X+X1:Y=Y+4:X2=X1:P=17408+X+Y*40
2020 H=PEEK(P):H1=PEEK(P+40):H2=PEEK(P+4
0+X1)
2030 IFX2=-1POKEP,153:POKEP+40,159:POKEP
+39,148 ELSE POKEP,152:POKEP+40,151:POKEP
+41,146:X2=1
2040 POKEP-21504,7:POKEP-21464,8:IFH2=32
POKEP-21464+X2,0
2050 SOUND9,15:SOUND10,15:SOUND3,2:SOUND
5,2:FORM=0TO255STEP4:SOUND2,M:FORN=1TO10:
NEXTN:SOUND4,255-M:FORN=1TO10:NEXTN,M:SOU
ND9,0:SOUND10,0
2060 GOTO650
2070 PRINT" █    █  █████  █   █  █████
 █  █   █"
2080 PRINT" █    █  █      █  █     █
 █  █  █ "
2090 PRINT" ██████  ████   ███      █
 █  ███  "
2100 PRINT" █    █  █      █  █     █
 █  █  █ "
2110 PRINT" █    █  █████  █   █    █
 █  █   █"
2120 FORL=17408TO17607:IFPEEK(L)=202 POK
EL,160
2130 NEXTL
2140 FORX=0TO38STEP2:PRINT@280+X,CHR$(12
9);CHR$(137);:NEXTX
2150 COLOUR16
2160 PRINT:PRINT
2170 PRINT"  Geschrieben von Juergen Buc
hmueller"
2180 PRINT"  fuer Trommeschlaeger Comput
er Studio"
2190 PRINT:PRINT"         Copyright (C)
1983"
2200 PRINT:PRINT
2210 PRINT"  REKORDPUNKTZAHL:";R2;"0"
2220 PRINT"  ─────────────────"
2230 PRINT"  LETZTER SPIELER:";P2;"0"
2240 PRINT"  ─────────────────"
2250 PRINT:PRINT"Druecken Sie (S) um das
 Spiel zu starten";
2260 PRINT"    oder (R) fuer die Rekordl
iste.";
```

```
2270 FORL=0TO7
2280 POKEL-2816,PEEK(((L+C)AND7)-3064)
2290 NEXTL
2300 C=C+1
2310 FORM=1TO500:NEXTM
2320 C$=INKEY$
2330 IFC$="S"RETURN
2340 IFC$<>"R"GOTO2270
2350 CLS:SOUND7,7:SOUND8,16:SOUND9,16:SO
UND10,16:SOUND6,31:SOUND12,50:SOUND13,9
2360 POKE17151,140:POKE17147,19:POKE1714
4,17:LGR
2370 PRINT@12," REKORDLISTE ":PRINT@52,"
 ____________________ ";
2380 FORL=0TO8:COLOURL+1
2390 PRINT@128+L*40,CHR$(49+L);".   ";
2400 FORM=31100+L*10TO31107+L*10
2410 IFPEEK(M)<32POKEM,32
2420 PRINTCHR$(PEEK(M));
2430 NEXTM
2440 PRINT"    ";
2450 PRINTPEEK(31108+L*10)+PEEK(31109+L*
10)*256;"0"
2460 NEXTL
2470 PRINT@530,"Druecken Sie (S).";
2480 IFPEEK(&HF804)<>8GOTO2480
2490 POKE17151,70:POKE17147,38:POKE17144
,32:LGR
2500 RETURN
2510 P1=P:FORM=0TO7:POKEP+M,32:NEXTM
2520 Y$=INKEY$:IFY$=""GOTO2520
2530 A=ASC(Y$):IFA=13PRINTCHR$(15);:RETU
RN
2540 IFA=8IFP>P1POKEP,32:P=P-1:PRINTCHR$
(8);".";CHR$(24);:GOTO2520
2550 IFA<32GOTO2520ELSEIFA>127GOTO2520
2560 IFP<P1+8POKEP,A:P=P+1:PRINTY$;
2570 GOTO2520
2580 FORL=17408TO18431:POKEL,32:NEXTL
2590 POKE&H4020,0:POKE&H4021,68
2600 RETURN
```

Die Zeichen von HEKTIK

CHR$(128) CHR$(129) CHR$(130) CHR$(131) CHR$(132) CHR$(133) CHR$(134)

CHR$(135) CHR$(136) CHR$(137) CHR$(138) CHR$(139) CHR$(140) CHR$(141)

CHR$(142) CHR$(143) CHR$(144) CHR$(145) CHR$(146) CHR$(147) CHR$(148)

CHR$(149) CHR$(150) CHR$(151) CHR$(152) CHR$(153) CHR$(154) CHR$(155)

CHR$(156) CHR$(157) CHR$(158) CHR$(159) CHR$(160) CHR$(161) CHR$(162)

Erläuterung des Programmes

Zeile(n)	Funktion/Bedeutung
10...320	DATA-Zeilen für die definierbaren Zeichen. Der Colour-Compiler legt DATAs als Doppelbytes ab Adresse 4804H (= 18436 dez.) ab.
330	Leseschleife für definierbare Zeichen. Kopiert 256 Bytes von 4804H nach F400H.
340	CHAR 2, also sind die Zeichen 128...191 selbstdefinierbar und 192...255 vorgegeben.
350	DATA-Zeilen für die HEKTIK-Melodie.
360	Melodie in Buffer nach 30720 kopieren (Adresse von A$ beim Colour-Compiler)
370	Cursor umdefinieren. Sechste und siebte Zeile sind gesetzt und nicht-blinkend.
380	Bildschirm und Farbspeicher löschen.
390	E = Anzahl der Verfolger -1: M2 = Anzahl der Jäger : R1 = Runde
400	Unterprogramm für Ausgabe des Titelbildes aufrufen und auf "S" warten.
410	B2 = Bonus : P2 = Punkte : Q & Q1 = Zähler
420	CLS mit POKEs aufrufen. Y- und X-Schleife für Zeichnen des Bodens eröffnen.
430	Boden besteht abwechselnd aus den Zeichen 129 und 137.
440	Farbe POKEn. 6 = COLOUR 5 = orange.
450	Nächster Ziegel, Ebene.
460	Unterste Zeile wird mit 202 belegt, damit hier das Hacken unmöglich ist.
470	Schleife für Anfangspositionen der Leitern.
480	A und B sind die X-Positionen der jeweils zwei Leitern in einer Etage.

Zeile(n)	Funktion/Bedeutung
490	Zweite Y-Schleife für Länge der Leitern.
500	Sprosse und Farbe (3 = weiß) der ersten Leiter in Bildschirm bzw. Farbspeicher poken.
510	Ebenso mit der zweiten Leiter verfahren.
520	Nächste Sprosse, Etage
530	mit COLOUR 11 in der letzten Zeile anzeigen.
540	Anzahl der restlichen Jäger rechts unten anzeigen
550	Wenn T = 0 (d.h. Alle Verfolger sind abgestürzt) dann Bonus zur Punktzahl addieren.
560	Soundregister vorbelegen.
570	T = Zahl der abgestürzten Verfolger : T2 = Anfang der Melodie -1 : O2 = Oktave : B2 = Bonus
580	Schleife für Vorbelegung der Positionen und Richtungen der Verfolger eröffnen.
590	X- und Y-Position poken.
600	X-Richtung poken (entspr. -1).
610	Y-Richtung poken (entspr. 0).
620	Zeichen unter Verfolger mit 32 (= Space) vorbelegen. Dieses Zeichen wird später anstelle des jeweiligen Verfolgers ausgegeben. Deshalb werden Leitern und Ziegel nicht zerstört.
630	Nächster Verfolger.
640	X = X-Position des Jägers : Y = Y-Position des Jägers : H, H1, H2 = Zeichen unter dem dem Jäger mit 32 (= Space) vorbelegen. : X2 = Richtung des Jägers für die Berechnung der Zeichen vorbelegen. : Y2 = Y-Richtung vorbelegen.
650	Tastaturabfrage: Bei A = 6 ist (BREAK) & (CLEAR) gedrückt.
660	X- und Y-Richtung gleich null setzen.

Zeile(n)	Funktion/Bedeutung
670	bei A = 32 ist Bit 5 gesetzt. Linkspfeil.
680	bei A = 64 ist Bit 6 gesetzt. Rechtspfeil.
690	bei A = 8 ist Bit 3 gesetzt. Pfeil nach oben.
700	bei A = 16 ist Bit 4 gesetzt. Pfeil nach unten.
710	P = Bildschirmadresse des Jägers.
720	I1 = Zähler für die blinkenden Bonus-Kugeln. Wenn I1 = 0, werden die beiden folgenden Zeilen übersprungen.
730	H1 = Zeichen das vorher unter den Beinen des Jägers stand.
740	H2 = Zeichen unter der Hacke.
750	C = Zähler : C1 ist abwechselnd 0 oder 1.
760	Wenn das Zeichen unter dem Kopf des Jägers gleich einem Verfolger ist, dann EXITUS.
770	Ebenso bei dem Zeichen unter der Hacke.
780	Wenn der Bonus noch größer Null ist, dann dekrementieren.
790	Wenn A AND 127 = 0, bedeutet dies, keine Pfeiltaste gedrückt wurde. Also wird bei einer Bewegung der letzte Jäger im Bildschirm gelöscht. Der Farbspeicher wird auf 15 (= weiß) gesetzt.
800	Wenn X-Position plus Richtung auf den rechten oder linken Rand kommen, dann X-Richtung Null setzen.
810...820	Übernahme der Richtungen in Variablen zur Berechnung der Zeichen für den Jäger.
830	P1 = Position + Y-Richtung P = Position + X-Richtung
840	Z = Zeichen unter den 'Füßen' des Jägers. Wenn alle Vergleiche nicht zutreffen, steht der Jäger nicht auf etwas, auf dem er laufen kann.

Zeile(N)	Funktion/Bedeutung
850	Wenn Z=135 oder Z=143, Absturz des Jägers.
860	Wenn der Jäger nicht auf einer Leiter steht, dann wird die Y-Richtung gleich Null gesetzt.
870	X,Y und P neu berechnen, wenn A AND 127 <> 0, dann neue Zeichen unter dem Jäger in H, H1 und H2 abspeichern.
880	Wenn I1 ≤ 1, dann blinken die Bonuskugeln nicht.
890	Wenn Zeichen unter der Hacke ein Zeichen der Bonuskugel ist, dann Unterprogramm für Bonusaddition ab Zeile 1880 aufrufen.
900	Farben für Jäger POKEn. (P-21504 ergibt Farbspeicheradresse, z.B. 17408-21504=-4096)
910	Wenn seitliche Bewegung und das Zeichen unter der Hacke ein Leerzeichen ist, dann Farbspeicher auf 0 (=grau) setzen.
920...940	In Abhängigkeit von den Richtungen werden die Zeichen des Jägers berechnet und in den Bildschirmspeicher gePOKEt.
950	A AND 128 = 128 bedeutet, daß die Leertaste gedrückt wurde (Hacken). Sonst nach Zeile 1070.
960	Wenn X-Richtung Null ist, kann nicht gehackt werden.
970	Ton beim Hacken.
980	Hacke auf und ab bewegen (Mit C1).
990	Wenn C1=0, dann ist die Hacke oben, und das Loch wird nicht vergrößert.
1000	P wird nun die Bildschirmposition unter der Hacke. Z speichert das dortige Zeichen.
1010	SHIFT gedrückt ? Wenn ja, dann 1050.
1020...1040	In Abhängigkeit vom Ziegeltyp das Loch vergrößern.
1050,1060	In Abhängigkeit vom Ziegeltyp das Loch schließen.

Zeile(n)	Funktion/Bedeutung
1070	In Abhangigkeit von der Spielrunde, wird unterschiedlich oft die Routine zur Bewegung der Verfolger aufgerufen. Ansonsten wird eine Warteschleife ausgefuhrt.
1080	Wenn Z2=1 ist, wurde man von einem Verfolger erwischt (siehe Unterprogramm ab 1400).
1090	Ausgabe von "** HEKTIK **", Punktzahl und Bonus in zufalligen Farben.
1100	Wenn alle Verfolger abgesturzt sind, wird Absturzzahler T gleich Null gesetzt, die Rundenzahl R1 um 1 erhoht und, wenn E<5 (E=Anzahl der Verfolger-1) wird E ebenfalls um 1 erhoht. Dann in die nachste Runde (Zeile 420).
1110	Zur Hauptschleife (Tastaturabfrage...).
1120...1140	Ton bei Verlust eines Jagers.
1150	M2 (Jagerzahl) erniedrigen, wenn M2=0 dann Spielende (Zeile 1180 folgende).
1160,1170	Bonus=0, zur Hauptschleife.
1180...1230	Erreichte Punktzahl wird mit der Rekordliste verglichen. Ist man nicht unter den ersten Neun, Sprung zum Titel.
1240...1390	Eingabe des Namens und Einsortieren in die Rekordliste.
1400	Schleife fur Verfolgerbewegung eroffnen.
1410	X,Y und Zeichen unter dem jeweiligen Verfolger Q aus dem Speicher in Variablen holen.
1420	Bei V=255 ist der Verfolger schon abgesturzt.
1430	Wenn Q ein Teil des Jagers ist, dann Flag Z2 setzen und Rucksprung.
1440	Wenn Q ein Teil der Bonuskugel ist, aber der Bonus nicht blinkt, wird Q=32 (Leerzeichen) gesetzt.
1450	X- u. Y-Richtung in Variablen speichern.

Zeile(n)	Funktion/Bedeutung
1460	Das letzte Zeichen unter dem jeweiligen Verfolger (= Q) wird wieder in den Bildschirmspeicher gePOKEt.
1470	Farbspeicher weiß setzen.
1480	Wenn Q ungleich Leiter (=128) und ungleich Leerzeichen (=32), dann Farbspeicher auf orange (=6).
1490	Wenn X-Position plus X-Richtung gleich linker oder rechter Rand, dann X-Richtung umdrehen.
1500	A wird Wert des Zeichens unterhalb und B oberhalb der Bildschirmposition des Verfolgers.
1510	Wenn Y-Position = 1 (oberste Etage) oder Y-Position = 21 (unterste Etage), dann Y-Richtung entsprechend umdrehen.
1520	Wenn ein Loch im Boden unter dem Verfolger ist (bei A=135 oder A=143), dann stürzt er ab.
1530	Wenn Verfolger auf Leiter, dann X-Richtung auf Null setzen oder wenn er direkt über dem Boden und außerdem auf dem Weg nach unten ist.
1540	Sonst X-Richtung mit 1 oder 3 belegen. (RND (2) OR 1 ergibt immer 1 oder 3).
1550	Wenn Verfolger auf dem Weg nach oben und auf einer Leiter, dann X-Richtung gleich Null setzen.
1560	Sonst wie bei Zeile 1540.
1570	X- und Y-Richtung zu den Positionen addieren.
1580	Neues Q aus dem Bildschirmspeicher in die Tabelle speichern.
1590	Auf die Bildschirmposition einen der beiden Verfolgertypen POKEn.
1600	Farbspeicher mit grün setzen (= 5).
1610	Neue X- und Y-Position in der Positionstabelle abspeichern.

Zeile(n)	Funktion/Bedeutung
1620	Wenn der Verfolger nicht uber seinesgleichen steht, dann Sprung zu 1660
1630...1650	Sonst Tabelle nach Verfolger mit gleicher X- und Y-Position durchsuchen, und wenn gefunden, dann dessen Wert fur Q ubernehemen.
1660	Nachster Verfolger.
1670	Zahler fur Noten um eins erhohen. Wenn Ende der Melodie erreicht ist, dann Zahler zurucksetzen und Zahler fur Oktave um eins erhohen. Wenn Oktave = 7 dann wieder bei 3 beginnen.
1680	A = Ton aus Tabelle. Mit PLAY auf 3. Tonkanal ausgeben. Wenn A<>0 dann Hullkurve auf Impuls programmieren.
1690	Wenn Bonuskugel sich schon bewegt (Flag gesetzt), dann zu Zeile 1840
1700...1730	Sonst mit einer mit der Rundenzahl zunehmender Wahrscheinlichkeit das Flag und den Zahler fur den Bonusblinker belegen.
1740	Absturz eines Verfolgers. Mit steigender Anzahl der Runden immer ofter das Loch schließen.
1750...1830	Absturz eines Verfolgers. Dabei werden die Zeichen und die Farbe unter dem absturzenden Verfolger immer in Q und Q1 zwischengespeichert. Anschließend wird zur Punktzahl Anzahl der Feinde mal 5 addiert.
1840	Wenn Zahler fur Bonuskugel (I1) noch großer Null ist, dann wird er um eins erniedrigt.
1850	Wenn I1 anschließend gleich Null ist, dann wird der Blinker geloscht und das Flag zuruckgesetzt.
1860	Sonst wird abwechselnd (AND 1) eine der zwei verschiedenen Kugeltypen ausgegeben.
1870	Zuruck zur Hauptschleife.
1880	Bonuskugel wurde beruhrt. Dann wird das Flag zuruckgesetzt und statt der Kugel "1000" angezeigt.

Zeile(n)	Funktion/Bemerkung
1890	Die Melodie bei einem Bonus wird mit Hilfe von CHR$(X) in B$ zwischengespeichert.
1900	Schleife für neun Töne. C wird aktueller Ton.
1910	Auf Kanal 3 mit Lautstärke 14 ausgeben.
1920	Warteschleife und nächster Ton.
1930	Die "1000" wieder löschen
1940	Sämtliche Zwischenspeicher für den 'Untergrund' des Jägers durch 32 (Leerzeichen) ersetzen. 100 zu den Punkten addieren (wird als 1000 angezeigt) und Zähler und Flag zurücksetzen.
1950	Zurück zur Hauptschleife.
1960	Das Loch, über dem der Verfolger, steht wieder schließen.
1970	Ton ausgeben.
1980	Zurück, kein Absturz.
1990	Der Jäger fällt eine Etage tiefer. Wenn die Hacke vorher in einer Leiter war, dann das Zeichen durch ein Leerzeichen ersetzen.
2000	Jäger auf alter Position mit den zwischengespeicherten Werten für den Untergrund löschen.
2010	Neue Position berechnen.
2020	Neue Hintergrundwerte aus dem Bildschirmspeicher lesen und in H, H1 und H2 ablegen.
2030	Aus der letzten X-Richtung die neuen Zeichen für den Jäger berechnen und in den Bildschirmspeicher POKEn.
2040	Farben für den Jäger in den Farbspeicher POKEn.
2050	Ton erzeugen und wieder abschalten.
2060	Zurück zur Hauptschleife (Tastaturabfrage).

Zeile(n)	Funktion/Bemerkung
2070...2110	Ausgabe des Titels.
2120,2130	CHR$(202) Grafik-Zeichen durch CHR$(160) ersetzen. Dieser wird dann anschließend umdefiniert, sodaß die Mauer durch die Schrift zu laufen scheint.
2140	Darunter eine Zeile Boden ausgeben.
2150...2260	Restliche Texte ausgeben (In COLOUR 16 = weiß).
2270...2290	Diese Routine kopiert in CHR$(160) die Mauer um jeweils ein Byte weitergeschoben.
2300	Zähler erhöhen.
2310	Warteschleife.
2320...2340	Tastaturabfrage auf "S" oder "R".
2350	Bei "R" wird die Rekordtabelle ausgegenben. Explosionsgeräusch in PSG programmieren.
2360	CRTC auf doppelte Zeichenhöhe umprogrammieren.
2370...2470	Rekordliste ausdrucken. Neun Plätze in verschiedenen Farben ausgeben. Hinter der Punktzahl eine zusätzliche Null anzeigen.
2480	Ist "S" gedrückt ? Nein, dann wieder abfragen.
2490	Zeichenhöhe wieder auf normal programmieren.
2500	Zurück vom Unterprogramm.
2510	Unterprogramm für Eingabe eines Namens mit 8 Buchstaben. Die RAM-Adresse des Buffers steht in der Variablen P. Zuerst den Buffer löschen.
2520...2570	Tastendrücke abspeichern bis RETURN gedrückt wird. Mehr als 8 Zeichen werden nicht angenommen und man kann mit Linkspfeil nicht vor die erste Bufferadresse P1 zurückgehen.
2580...2600	Schneller CLS, der Leerzeichen in den Bildschirmspeicher POKEt. Die Cursorposition in 4020H und 4021H wird auf linke obere Ecke gePOKEt.

Bubble Sort

"Bubble Sort" ist ebenfalls ein Spielprogramm, daß Sie mit dem "Colour Compiler" compilieren müssen.
Sie können das Programm aber auch mal im normalen Basic laufen lassen, um die immense Geschwindigkeitserhöhung, die der Compiler bringt, zu sehen. Auf eine Erklärung des Programmes sei verzichtet - die Spielregeln finden Sie im Programm selbst.

Programmlisting:

```
10 CLS:COLOUR1
20 PRINT@14,"Bubble Sort"
30 PRINT:PRINT"Bei diesem Spiel geht es darum, die"
40 PRINT"eine Sorte Kugeln (";:COLOUR3:PRINTCHR$(230);:COLOUR1:PRINT")
 auf die eine und"
50 PRINT"die andere Sorte (";:COLOUR2:PRINTCHR$(235);:COLOUR1:PRINT")
auf die andere"
60 PRINT"Seite des Bildschirms zu bringen. Den"
70 PRINT"blauen Teil des Mittelstreifens koennen"
80 PRINT"Sie mit dem Hoch- und Runterpfeil be-"
90 PRINT"wegen und mit <RETURN> oeffnen.":PRINT:PRINT:PRINT"Bitte drue
cken Sie <RETURN>"
100 A=PEEK(&HF840)AND1
110 IFA=0THEN100
120 SOUND7,248:SOUND8,16:SOUND9,0:SOUND10,0:SOUND1,0:SOUND0,100:SOUND1
2,4
130 COLOUR4:CLS
140 '******** UMRANDUNG ZEICHNEN
150 FORX=0TO39
160 POKEX-4096,3:POKE17408+X,202
170 POKEX-3216,3:POKE18288+X,202
180 NEXTX
190 FORY=40TO840STEP40
200 POKEY-4096,3:POKE17408+Y,202
210 POKEY-4057,3:POKE17447+Y,202
220 POKEY-4076,6:POKE17428+Y,202
230 NEXTY
240 '******** KUGELN ZUFAELLIG SETZEN
250 FORL=0TO9
260 R=2
270 X=RND(38):Y=RND(21):A=(LAND1)*3+R:X1=RND(2)OR1:Y1=RND(2)OR1
280 IFX=20GOTO270
290 POKE31000+L,X:POKE31010+L,Y:POKE31020+L,A:POKE31030+L,X1:POKE31040
+L,Y1
300 GOSUB760
310 NEXTL
320 Z=0
330 P1=40:P=40:D1=0
340 D1=D1+1:PRINT@920,D1;" Zeiteinheiten"; :A=PEEK(&HF840)
350 IF(AAND8)=8IFP>40LETP=P-40
```

```
360 IF(AAND16)=16IFP<800LETP=P+40
370 FORY=40TO840STEP40:POKEY-4076,6:POKE17428+Y,202
380 IFY=PPOKEY-4076,8ELSEIFY=P+40POKEY-4076,8
390 IF(AAND1)=1IFY=PPOKE17428+Y,32ELSEIFY=P+40POKE17428+Y,32
400 NEXTY
410 GOSUB430
420 GOTO340
430 C1=0:C2=0:FORL=0TO9
440 X=PEEK(31000+L):Y=PEEK(31010+L)
450 A=PEEK(31020+L)
460 X1=PEEK(31030+L)-2:Y1=PEEK(31040+L)-2
470 IFX<20IF(LAND1)=0LETC1=C1+1
480 IFX>20IF(LAND1)=1LETC1=C1+1
490 IFX<20IF(LAND1)=1LETC2=C2+1
500 IFX>20IF(LAND1)=0LETC2=C2+1
510 POKE 17408+X+Y*40,32
520 B=PEEK(17408+X+X1+Y*40)
530 C=PEEK(17408+X+(Y+Y1)*40)
540 D=PEEK(17408+X+X1+(Y+Y1)*40)
550 SOUND1,1:IFD<>32IFB=32LETY1=0-Y1:SOUND13,9:GOTO590 ELSE IFC=32LETX
1=0-X1:SOUND13,9:GOTO590
560 SOUND1,0
570 IFB<>32LETX1=0-X1:SOUND13,9
580 IFC<>32LETY1=0-Y1:SOUND13,9
590 X=X+X1:Y=Y+Y1:GOSUB760
600 POKE31000+L,X
610 POKE31010+L,Y
620 POKE31030+L,X1+2
630 POKE31040+L,Y1+2
640 NEXTL
650 IFZ=0IFC1=10GOTO670ELSEIFC2=10GOTO670
660 RETURN
670 RESTORE:POKEP+17428,202:POKEP+17468,202:Z=1:FORQ=1TO255STEP6:GOSUB
430
680 SOUND9,15:SOUND10,15
690 SOUND3,0:SOUND2,Q:SOUND5,0:SOUND4,255-Q
700 NEXTQ
710 SOUND7,255
720 CLS:PRINT"Sie haben es in";D1:PRINT"Zeiteinheiten geschafft."
730 A=PEEK(&HF840)AND1
740 IFA=0THEN730
750 GOTO10
760 POKE &HF000+X+Y*40,A
770 IFA=2THENR1=230ELSER1=235
780 POKE &H4400+X+Y*40,R1
790 RETURN
```

Tilgungsplan

Das folgende Programm hilft Ihnen bei der Berechnung der Abzahlung einer Ratenschuld. Dabei kann man wählen, ob die Schuld mit konstanter Annuität (Gesamtbelastung = Zinsen + Tilgung) oder mit konstanter Tilgung erfolgen soll.

Programmlisting:

```
10 CLS
20 COLOUR3:PRINT"Tilgungsplan einer Ratenschuld"
30 PRINT
40 CLEAR
50 COLOUR4:INPUT"1 - Konstante Gesamtbelastung
2 - Konstante Tilgung";B
60 ONBGOTO70,220
70 INPUT"Annuitaet, Zins, Kapital eingeben
";A,P,K#
80 PRINT
90 COLOUR2:PRINT"Zeit        Zins    Tilgung Restschuld"
100 K1#=K#
110 X=X+1
120 ZI=INT(K#*P+.5)/100
130 T=A-ZI
140 IFT>=K#THENT=K#
150 R#=K#-T
160 D$=" ### #######.## #######.## #######.##"
170 PRINTUSINGD$;X;ZI;T;R#
180 IFT<=0THENCOLOUR7:PRINT"Eine Tilgung ist nicht moeglich":PRINT"Die
 Zinsen betragen";ZI;"DM und sind groesser":PRINT"als die Tilgung von"
;T;"DM":GOTO490
190 IFR#<=0THENPRINT"In";X;"Zeitperioden ist die Schuld von":PRINTK1#;
"DM getilgt":GOTO490
200 K#=R#
210 GOTO110
220 DIM Z(1000)
230 A$=" ### #######.## ######.## #######.##"
240 B$="K=#######.## N=### P=###.###"
250 PRINT
260 INPUT"Bitte Kapital, Zeit, Zinssatz eingeben
";K,N,P
270 K1=K
```

```
280 PRINT
290 PRINT
300 PRINT
310 PRINTUSINGB$;K;N;P
320 COLOUR2:PRINT"Zeit      Zinsen   Tilgung  Annuitaet"
330 PRINT
340 T=INT(100*K/N+.5)/100
350 FORI=1TON-1
360 IF(I=21)OR(I=41)OR(I=61)THENINPUT"Weiter";C$
370 Z=INT(K*P+.5)/100
380 PRINTUSINGA$;I;Z;T;T+Z
390 K=K-T
400 Z(I)=Z
410 NEXTI
420 U=INT(K*P+.5)/100
430 Z(N)=U
440 PRINTUSINGA$;N;U;K;U+K
450 FORI=1TON
460 Z1=Z1+Z(I)
470 NEXTI
480 PRINT"Gesamte Zahlungen =";Z1+K1
490 PRINT:COLOUR8:INPUT"Andere Berechnung (J/N)";C$
500 IFLEFT$(C$,1)="J"ORLEFT$(C$,1)="j"THEN10
510 END
```

Gleichungen mit 3 Unbekannten

Dieses Mathematikprogramm berechnet die 3 Unbekannten von 3 Gleichungen der Form ax+by+cz=d.

Programmlisting:

```
10 CLS:COLOUR1
20 PRINT"Dieses Programm berechnet aus drei ":PRINT"Gleichungen der Fo
rm"
30 PRINT"ax+by+cz=d (a,b,c und d gegeben) die":PRINT"Unbekannten x,y u
nd z"
40 PRINT
50 COLOUR2:INPUT"a1, b1, c1 und d1";A1,B1,C1,D1
60 COLOUR3:INPUT"a2, b2, c2 und d2";A2,B2,C2,D2
70 COLOUR7:INPUT"a3, b3, c3 und d3";A3,B3,C3,D3
80 D=A1*B2*C3+B1*C2*A3+C1*A2*B3-A3*B2*C1-B3*C2*A1-C3*A2*B1
90 DX=D1*B2*C3+B1*C2*D3+C1*D2*B3-D3*B2*C2-B3*C2*D1-C3*D2*B1
100 DY=A1*D2*C3+D1*C2*A3+C1*A2*D3-A3*D2*C1-D3*C2*A1-C3*A2*D1
110 DZ=A1*B2*D3+B1*D2*A3+D1*A2*B3-A3*B2*D1-B3*D2*A1-D3*A2*B1
120 PRINT:PRINT:COLOUR4
130 IFD=0THENPRINT"Das System ist nicht loesbar.":END
140 PRINT"x =";DX/D
150 PRINT"y =";DY/D
160 PRINT"z =";DZ/D
170 END
```

Pascalsches Dreieck

Für dieses Programm wird ein Drucker benötigt. Ausgedruckt wird das sogenannte Pascalsche Dreieck (siehe Ausdruck unten).
Das Pascalsche Dreick wird in der Mathematik dazu benutzt, um die Binominalkoeffizienten von Gleichungen n-ten Grades abzulesen.

```
                    Pascalsches Dreieck
n= 0---------------------------------1--------------------------------
n= 1------------------------------1     1-----------------------------
n= 2---------------------------1     2     1--------------------------
n= 3------------------------1     3     3     1-----------------------
n= 4---------------------1     4     6     4     1--------------------
n= 5------------------1     5    10    10     5     1-----------------
n= 6---------------1     6    15    20    15     6     1--------------
n= 7------------1     7    21    35    35    21     7     1-----------
n= 8---------1     8    28    56    70    56    28     8     1--------
n= 9------1     9    36    84   126   126    84    36     9     1-----
n=10---1    10    45   120   210   252   210   120    45    10     1--
```

Programmlisting:

```
10 LPRINTCHR$(15)
20 CLEAR100
30 DIMDR(17,17)
40 LPRINTCHR$(14)"            Pascalsches Dreieck"
50 LPRINTCHR$(15)
60 DR(0,0)=1
70 CLS
80 PRINT"Pascalsches Dreieck"
90 TAB=32
100 FORANZAHL=1TO11
110 LPRINT"n=";
120 LPRINTUSING"##";ANZAHL-1;
130 FORA=0TOTABB
140 LPRINT"-";
150 NEXT
160 FORS=1TOANZAHL
170 DR(ANZAHL,S)=DR(ANZAHL-1,S-1)+DR(ANZAHL-1,S)
180 IFS>1THENLPRINTUSING"######";DR(ANZAHL,S);:NEXTELSELPRINTUSING"#":
DR(ANZAHL,S);:NEXT
190 FORA=1TOTAB
200 LPRINT"-";
210 NEXT
220 LPRINT
230 TAB=TAB-3
240 NEXT
250 LPRINTSTRING$(6,10)
```

Garbage Collect

Das Programmlisting, das nun folgt, soll Ihnen einen Effekt verdeutlichen, dem Sie möglicherweise schon begegnet sind, ohne ihn sich erklären zu können.
Gemeint ist, daß sich der Computer in längeren Basic-Programmen für einige Sekunden "aufhängt", d.h. daß das Programm stehenbleibt und sogar mit der BREAK-Taste nichts mehr zu machen ist.
Wenn Sie das folgende Programm eingeben und starten, wird das Problem sehr deutlich: Obwohl zwischen den einzelnen PRINTs nur ganz normale Variablenzuweisungen ausgeführt werden, dauert dies bis zu einigen Sekunden. Dies kann auch dann Probleme bereiten, wenn Sie Daten vom Band einlesen, da durch eine derartige Verzögerung Daten verlorengehen können.
Was ist nun der Grund für diese Verzögerung und wie kann man das Problem umgehen und seine Basicprogramme damit schneller machen?
Der Grund ist Zeile 50:
50 DIM A(MEM/3-500)
In dieser Zeile wird ein großes Feld dimensioniert (abhängig vom vorhandenen Speicher). Wird nun eine neue Variable eingeführt, wird das ganze Feld verschoben, um für die Variable Platz im Speicher zu schaffen. Dies verursacht besagte Verzögerungen. Fügt man nun eine Zeile ein, die alle später gebrauchten Variablen bereits vor dem DIM-Befehl schon einmal benutzt, werden diese schon vor dem Feld im Speicher angelegt, das dann später nicht mehr verschoben werden muß. Dies können Sie z.B. bei dem vorliegenden Programm einfach dadurch erreichen, indem Sie die ganz unten gelistete Zeile 5 mit eingeben, die allen benutzten Variablen den Wert 0 zuweist.
Das Programm läuft dann erheblich schneller!

Programmlisting:

```
10 CLS:COLOUR1
20 DEFSTRA
30 R=0
40 Q=R
50 DIMA(MEM/3-500)
60 A(0)="        "
70 A(1)="<RETURN>"
80 CLS
90 PRINT"Druecken Sie <RETURN> zur Demonstration":PRINT"des Garbage Co
llect"
100 Q=1-Q
```

```
110 COLOURRND(8):PRINT§13.A(Q);
120 FORR=1TO20
130 IFINKEY$<>CHR$(13)THENNEXT:GOTO100
140 CLS:COLOUR9
150 PRINT"X =";
160 X=2
170 PRINTX
180 X1=X
190 PRINT"Zwei und zwei sind";
200 Y=X+X
210 PRINTY
220 Y1=Y
230 COLOUR1:PRINT"Wussten ";
240 A=""
250 COLOUR2:PRINT"Sie";
260 B=0
270 COLOUR3:PRINT", dass ";
280 C=0
290 COLOUR4:PRINT"Ihr ";
300 D=0
310 COLOUR5:PRINT"Computer ";
320 E=0
330 COLOUR6:PRINT"so "
340 F=0
350 COLOUR7:PRINT"lang";
360 G=0
370 PRINT"s";
380 H=0
390 PRINT"a";
400 I=J
410 J=K
420 PRINT"m ";
430 K=C
440 L=K
450 M=B
460 COLOUR8:PRINT"ist?"
470 COLOUR1
```

```
5 B=0:C=B:D=B:E=B:F=B:G=B:H=B:I=B:J=B:K=B:L=B:M=B:Q=B:R=B:X=B:X1=B:Y=B
:Y1=B:A$=""
```

Ein Maschinensprache-Monitor in Basic

Da es im Laufe dieses Buches nötig sein wird, kleinere Programme in hexadezimal einzugeben oder sich Speicherinhalte anzusehen, geben Sie bitte das hier aufgelistete Basicprogramm ein und speichern Sie es auf Kassette:

```
10 CLS
20 CLEAR 1000
30 PRINT"Command";
40 INPUT A$
50 IF A$="" THEN 30
60 GOSUB 1460
70 IF B$="P" THEN PR=1: GOSUB 1460 ELSE PR=0
80 C$="HAETGRW"
90 FOR A=1 TO LEN(C$)
100 IF B$<>MID$(C$, A, 1)THEN NEXT A: GOTO 30
110 ON A GOTO 130, 390, 590, 740, 900, 1040, 1730
120 GOTO 30
130 GOSUB 1510
140 X=Z
150 Z=X
160 GOSUB 1600
170 PRINT Z$;" ";
180 IF PR=1 THEN LPRINT Z$;" ";
190 C=0
200 FOR Y=0 TO 7
210 Z=PEEK(X+Y)
220 C=C+Z
230 GOSUB 1600
240 PRINT RIGHT$(Z$, 2);" ";
250 IF PR=1 THEN LPRINT RIGHT$(Z$, 2);" ";
260 NEXT Y
270 PRINT"  ";
280 IF PR=1 THEN LPRINT"  ";
290 Z=C
300 GOSUB 1600
310 PRINT RIGHT$(Z$, 2)
320 IF PR=1 THEN LPRINT RIGHT$(Z$, 2)
330 X=X+8
340 KB=PEEK(&HF840)
350 IF KB=2 THEN 30
360 IF KB<>128 THEN 150
370 KB=PEEK(&HF840)
380 IF KB<>1 THEN 370 ELSE 150
390 GOSUB 1510
400 X=Z
410 Z=X
420 GOSUB 1600
430 PRINT Z$;" ";
440 IF PR=1 THEN LPRINT Z$;" ";
450 FOR Y=0 TO 23
460 Z=PEEK(X+Y)
470 IF Z<32 THEN Z=ASC(".")
480 PRINT CHR$(Z);
490 IF PR=1 THEN LPRINT CHR$(Z);
500 NEXT Y
510 PRINT
```

```
520 IF PR=1 THEN LPRINT
530 X=X+24
540 KB=PEEK(&HF840)
550 IF KB=2 THEN 30
560 IF KB<>128 THEN 410
570 KB=PEEK(&HF840)
580 IF KB<>1 THEN 570 ELSE 410
590 GOSUB 1510
600 X=Z
610 Z=X
620 GOSUB 1600
630 PRINT Z$;" - ";
640 Z=PEEK(X)
650 GOSUB 1600
660 PRINT RIGHT$(Z$, 2);" - ";
670 A$=""
680 INPUT A$
690 IF A$="" THEN 30
700 GOSUB 1510
710 POKE X, Z AND 255
720 X=X+1
730 GOTO 610
740 GOSUB 1510
750 X=Z
760 Z=X
770 GOSUB 1600
780 PRINT Z$;" - ";
790 Z=PEEK(X)
800 GOSUB 1600
810 PRINT RIGHT$(Z$, 2);" (";
820 IF Z<32 THEN Z=ASC(".")
830 PRINT CHR$(Z);") - ";
840 A$=""
850 INPUT A$
860 IF A$="" THEN 30
870 POKE X, ASC(A$)
880 X=X+1
890 GOTO 760
900 GOSUB 1510
910 Z1=INT(Z/256)AND 255
920 Z2=Z AND 255
930 POKE 16526, Z2
940 POKE 16527, Z1
950 GOSUB 1460
960 IF B$=" " THEN 950
970 A$=B$+A$
980 GOSUB 1510
990 Z=Z AND 255
1000 Z=USR(Z)
1010 GOSUB 1600
1020 PRINT RIGHT$(Z$, 2)
1030 GOTO 30
1040 GOSUB 1510
```

```
1050 IF Z>=0 THEN X1=Z ELSE X1=Z+65536
1060 GOSUB 1460
1070 IF B$=" " THEN 1060
1080 A$=B$+A$
1090 GOSUB 1510
1100 IF Z>=0 THEN X2=Z ELSE X2=Z+65536
1110 PRINT"  A      B     A+B    A-B"
1120 PRINT" ";
1130 Z=X1
1140 GOSUB 1600
1150 PRINT Z$;"  ";
1160 Z=X2
1170 GOSUB 1600
1180 PRINT Z$;"  ";
1190 Z=X1+X2
1200 IF Z>65535 THEN Z=Z-65536
1210 GOSUB 1600
1220 PRINT Z$;"  ";
1230 Z=X1-X2
1240 IF Z<0 THEN Z=Z+65536
1250 GOSUB 1600
1260 PRINT Z$
1270 Z=X1
1280 GOSUB 1420
1290 PRINT Z$;" ";
1300 Z=X2
1310 GOSUB 1420
1320 PRINT Z$;" ";
1330 Z=X1+X2
1340 IF Z>65535 THEN Z=Z-65536
1350 GOSUB 1420
1360 PRINT Z$;" ";
1370 Z=X1-X2
1380 IF Z<0 THEN Z=Z+65536
1390 GOSUB 1420
1400 PRINT Z$
1410 GOTO 30
1420 Z$=STR$(Z)
1430 Z$=RIGHT$(Z$, LEN(Z$)-1)
1440 IF LEN(Z$)<5 THEN Z$=STRING$(5-LEN(Z$), 32)+Z$
1450 RETURN
1460 B$=""
1470 IF LEN(A$)=0 THEN RETURN
1480 B$=LEFT$(A$, 1)
1490 A$=RIGHT$(A$, LEN(A$)-1)
1500 RETURN
1510 Z=0
1520 GOSUB 1460
1530 IF B$<"0" OR B$>"F" THEN 1580
1540 IF B$>"9" AND B$<"A" THEN 1580
1550 IF B$<="9" THEN Z1=ASC(B$)-48 ELSE Z1=ASC(B$)-55
1560 Z=Z*16+Z1
1570 GOTO 1520
```

```
1580 IF Z>32767 THEN Z=Z-65536
1590 RETURN
1600 IF Z>32767 THEN Z=Z-65536
1610 Z1=INT(Z/256)AND 255
1620 Z2=Z AND 255
1630 Z$=""
1640 Z3=INT(Z1/16)
1650 GOSUB 1710
1660 Z3=Z1 AND 15
1670 GOSUB 1710
1680 Z3=INT(Z2/16)
1690 GOSUB 1710
1700 Z3=Z2 AND 15
1710 IF Z3<10 THEN Z$=Z$+CHR$(Z3+48) ELSE Z$=Z$+CHR$(Z3+55)
1720 RETURN
1730 RESTORE
1740 WR$=""
1750 FOR X=1 TO 127
1760 READ A
1770 WR$=WR$+CHR$(A)
1780 NEXT
1790 X=VARPTR(WR$)
1800 X=PEEK(X+1)+256*PEEK(X+2)
1810 IF X>32767 THEN X=X-65536
1820 R=X+95
1830 R1=INT(R/256)AND 255
1840 R2=R AND 255
1850 POKE X+66, R2
1860 POKE X+67, R1
1870 POKE X+75, R2
1880 POKE X+76, R1
1890 GOSUB 2140
1900 GOSUB 2210
1910 GOSUB 2140
1920 GOSUB 2210
1930 GOSUB 2140
1940 PRINT"Filename";
1950 INPUT PN$
1960 FOR X1=X TO X+5
1970 POKE X1, 32
1980 NEXT
1990 FOR X1=1 TO LEN(PN$)
2000 POKE X-1+X1, ASC(MID$(PN$, X1, 1))
2010 NEXT
2020 X=X+6
2030 X1=INT(X/256)AND 255
2040 X2=X AND 255
2050 POKE 16526, X2
2060 POKE 16527, X1
2070 X1=USR(0)
2080 GOTO 30
```

```
2090 DATA 0, 0, 0, 0, 0, 0, 32, 32, 32, 32, 32, 32
2100 DATA 42, 142, 64, 17, 250, 255, 25, 205, 63, 2, 62, 85, 205, 31,
2, 6, 6, 126, 35, 205, 31, 2, 16, 249, 17, 244, 255, 25, 229, 221, 225
, 221, 94, 0, 221, 86, 1, 221, 110, 2, 221, 102, 3
2110 DATA 175, 237, 82, 35, 124, 183, 40, 8, 6, 0, 205, 0, 0, 37, 24,
244, 125, 69, 183, 196, 0, 0, 62, 120, 205, 31, 2
2120 DATA 221, 126, 4, 205, 31, 2, 221, 126, 5, 205, 31, 2, 201
2130 DATA 62, 60, 205, 31, 2, 120, 205, 31, 2, 123, 205, 31, 2, 79, 12
2, 205, 31, 2, 129, 79, 26, 19, 205, 31, 2, 129, 79, 16, 247, 195, 31,
 2
2140 GOSUB 1510
2150 Z1=INT(Z/256)AND 255
2160 Z2=Z AND 255
2170 POKE X, Z2
2180 POKE X+1, Z1
2190 X=X+2
2200 RETURN
2210 GOSUB 1460
2220 IF B$=" " THEN 2210
2230 A$=B$+A$
2240 RETURN
```

Dieser Monitor unterstuzt 7 Befehle, namlich:

1. Hex-Dump

   Eingabe: Hadresse

   Ab der eingegebenen Adresse wird der Speicherinhalt in hexadezimal gelistet. Jeweils nach Ausgabe einer Zeile wird die Tastatur auf folgende Tasten abgefragt:

   <CLEAR> - Listing abbrechen.

   <Leertaste> - Listing anhalten, bis <RETURN> gedruckt wird.

   Die einzelnen Zeilen haben folgendes Format:

   Die ersten 4 Zeichen geben die Adresse an, in der das erste der 8 Bytes, die in dieser Zeile gelistet werden, steht. Dann folgen die 8 Bytes und zum Schluß, etwas abgesetzt, folgt die sogenannte Checksum. Dieses Byte ist das niederwertige Byte der Summe aller 8 Bytes und dient als Prufmoglichkeit fur die Richtigkeit einer Eingabe. Alle Maschinenprogramme in diesem Buch werden als Hex-Dump mit Prufsumme gelistet, so daß Sie eine einfache Moglichkeit haben, die Richtigkeit Ihrer Eingabe zu prufen.

2. ASCII-Dump

   Eingabe: Aadresse

   Ab der eingegebenen Adresse wird der Speicherinhalt als ASCII-Zeichen gelistet. Jede Zeile beginnt mit der Adresse, in der das erste der 24 folgenden Zeichen steht, dann folgen die ASCII-Zeichen. Jedes Byte unter 32 dez. wird als '.' dargestellt, um die Bildschirmausgabe nicht durch Codes wie 'Backspace' etc. zu zerstoren. Auch beim ASCII-Dump kann die Ausgabe durch <CLEAR> abgebrochen und durch <Leertaste> angehalten werden.

3. Edit hexadezimal

   Eingabe: Eadresse

   Ab der eingegebenen Adresse kann der Speicherinhalt verandert werden. Es wird jeweils die Adresse und der bisherige Inhalt dieser Speicherstelle ausgegeben. Sie konnen dann ein neues Byte eingeben, das dann in dieser Speicherstelle abgelegt wird. Wenn Sie statt einer Eingabe nur <RETURN> drukken, wird der Editmodus abgebrochen.

4. Edit ASCII

   Eingabe: Tadresse

   Ab der eingegebenen Adresse konnen Sie Texte direkt in den Speicher eingeben. Die Adresse wird angegeben, ihr Inhalt wird in hexadezimal und ASCII (Code kleiner als 32 = '.') ausgegeben. Sie konnen nun ein Zeichen eingeben, das an dieser Speicherstelle gespeichert wird. Auch hier wird die Eingabe abgebrochen, wenn Sie nur <RETURN> drucken. Achtung: Um ein Leerzeichen in den Speicher einzugeben, mussen Sie folgendes eingeben: " ". Dies liegt daran, daß der INPUT-Befehl des Colour-Basic alle fuhrenden Leerzeichen ignoriert, eine Eingabe eines Leerzeichens allein also als Leerstring angesehen wird.

5. Go to memory
   Eingabe: Gadresse byte
   Es wird zu einem Programm gesprungen, das bei der eingegebenen Adresse beginnt. Wenn Ihr Programm mit einem CALL 0A7FH beginnt, so können Sie ein Byte in das L-Register übergeben. Wenn Sie Ihr Programm mit einem JP 0A9AH beenden, so wird der Inhalt des L-Registers wieder an das Basic übergeben und der Monitor gibt es in hexadezimal aus.
6. Hexadezimal Rechnung
   Eingabe: Azahl1 zahl2
   Zahl1, Zahl2, ihre Summe und ihre Differenz werden in hexadezimal und ASCII ausgegeben.
7. SYSTEM-Band schreiben
   Eingabe: Wstart ende einsprung
   Zusätzlich wird noch ein Programmname abgefragt. Dieser Name darf maximal 6 Zeichen lang sein und das erste Zeichen muß ein Buchstabe sein. Dann wird der Bereich von start bis ende im SYSTEM-Band-Format auf Kassette geschrieben. Um dieses Band wieder einzulesen, benutzen Sie den SYSTEM-Befehl des Colour-Basic und geben Sie den von Ihnen gewählten Programmnamen an.

Anmerkung: Für das H und das A-Kommando gilt:
PHadresse bzw. PAadresse gibt das entsprechende Listing parallel auf Bildschirm und Drucker aus.

Erklärung des Programms:

Zeile 10 - 20:	Initialisierung: Bildschirm löschen und Stringspace vergrößern.
Zeile 30 - 50:	Kommandoeingabe. Wenn nur <RETURN> gedrückt wurde, Eingabe wiederholen.
Zeile 60 - 70:	Erstes Zeichen der Eingabe lesen. Wenn P, Druckerflag (PR) setzen und nächstes Zeichen lesen.
Zeile 80 - 110:	Kommando aus C$ heraussuchen und die entsprechende Routine anspringen: H = 130 A = 390 E = 590 T = 740 G = 900 R = 1040 W = 1730
Zeile 120:	Falls Zeichen kein zulässiges Kommando, wieder zur Eingabe springen.

<u>Zeile 130 - 380: Hex-Dump</u>

Zeile 130 - 140:	Adresse aus A$ in Z lesen und in X umspeichern.
Zeile 150 - 160:	X in Hexadezimalzahl in Z$ umwandeln.
Zeile 170:	Adresse des ersten der folgenden 8 Bytes ausgeben.
Zeile 180:	Falls nötig, Ausgabe zusätzlich auf Drucker.
Zeile 190:	Prüfsumme auf Null initialisieren.
Zeile 200 - 260:	8 Bytes in hexadezimal auf Bildschirm (und auf Druckerm wenn nötig) ausgeben und Prüfsumme berechnen.
Zeile 270 - 320:	Prüfsumme ausgeben.
Zeile 330:	Startadresse um 8 erhöhen.
Zeile 340:	KB fragt eine Tastenreihe ab.
Zeile 350:	Wenn <CLEAR> gedrückt, Sprung zur Kommandoeingabe.
Zeile 360:	Wenn nicht <Leertaste> gedrückt, nächste Zeile ausgeben.
Zeile 370 - 380:	Warten, bis <RETURN> gedrückt, dann nächste Zeile ausgeben.

<u>Zeile 390 - 580: ASCII-Dump</u>

Zeile 390 - 400:	Adresse aus A$ in Z lesen und in X umspeichern.
Zeile 410 - 440:	Startadresse der Zeile ausgeben.
Zeile 450 - 500:	24 ASCII-Zeichen ausgeben.
Zeile 510 - 520:	Cursor an den Anfang der nächsten Zeile bringen.
Zeile 530:	Startadresse um 24 erhöhen.

Zeile 540 - 580:	Tastaturabfrage: <CLEAR> - Listing abbrechen <Leertaste> - Listing anhalten.

Zeile 590 - 730: Edit hexadezimal

Zeile 590 - 600:	Adresse aus A$ in Z lesen und in X umspeichern.
Zeile 610 - 630:	Adresse und Trennstrich ausgeben.
Zeile 640 - 660:	Bisherigen Inhalt der Speicherstelle X und zweiten Trennstrich ausgeben.
Zeile 670 - 680:	Eingabe in A$ lesen.
Zeile 690:	Wenn nur <RETURN> gedrückt wurde, Eingabe abbrechen.
Zeile 700 - 710:	Das eingegebene Byte in Adresse X ablegen. Das Z AND 255 dient dazu, sicherzustellen, daß nicht aus Versehen eine Zahl größer als 255 dez. gePOKEt wird.
Zeile 720 - 730:	Adresse um 1 erhöhen und Eingabevorgang wiederholen.

Zeile 740 - 890: Edit ASCII

Zeile 740 - 750:	Adresse aus A$ in Z lesen und in X umspeichern.
Zeile 760 - 780:	Adresse und ersten Trennstrich ausgeben.
Zeile 790 - 830:	Inhalt der Speicherstelle X in hexadezimal und als ASCII-Zeichen ausgeben.
Zeile 840 - 850:	Eingabe in A$ lesen.
Zeile 860:	Wenn nur <RETURN> gedrückt, Eingabe abbrechen.
Zeile 870:	ASCII-Wert des eingegebenen Zeichens in X ablegen.
Zeile 880:	Adresse um 1 erhöhen.
Zeile 890:	Eingabevorgang wiederholen.

Zeile 900 - 1030: Go to memory

Zeile 900:	Sprungadresse aus A$ in Z lesen.
Zeile 910 - 920:	Z1 = höherwertiges Byte von Z. Z2 = niederwertiges Byte von Z.
Zeile 930 - 940:	USR-Sprungadresse festlegen.
Zeile 950 - 970:	Alle Leerzeichen am Anfang von A$ wegstreichen.
Zeile 980 - 990:	Zu übergebendes Byte aus A$ in Z lesen.
Zeile 1000:	Programm starten. Das Argument in Z übergeben und das Ergebnis des Programms wieder in Z abspeichern.
Zeile 1010 - 1020:	Ergebnis in hexadezimal ausgeben.
Zeile 1030:	Sprung zur Kommandoeingabe.

Zeile 1040 - 1410: Hexadezimalrechnung

Zeile 1040: Zahl aus A$ in Z lesen.

Zeile 1050: Aus einer vorzeichenbehafteten Zahl in Z eine positive Zahl in X1 machen. (-32768 <= Z <= 32767; 0 <= X1 <= 65535)

Zeile 1060 - 1080: Alle Leerzeichen am Anfang von A$ löschen.

Zeile 1090: Zahl aus AS in Z lesen.

Zeile 1100: Aus einer vorzeichenbehafteten Zahl in Z eine positive Zahl in X2 machen (siehe 1050).

Zeile 1110: Kopfzeile der Tabelle ausgeben.

Zeile 1120 - 1260: X1, X2, X1 + X2 und X1 - X2 in hexadezimal ausgeben.

Zeile 1270 - 1400: X1, X2, X1 + X2 und X1 - X2 in dezimal ausgeben.

Zeile 1410: Sprung zur Kommandoeingabe

Zeile 1420 - 1450: Zahl in Z in Dezimalstring umwandeln, Z$ enthält Zahl rechtsbündig.

Zeile 1420: Z$ = Zahl aus Z in Stringform.

Zeile 1430: Vorzeichen aus Z$ entfernen.

Zeile 1440: Falls nötig, Z$ mit Leerzeichen auf 5 Zeichen auffüllen.

Zeile 1450: Rückkehr zum aufrufenden Programm.

Zeile 1460 - 1500: Das erste Zeichen aus A$ in B$ übertragen und A$ um ein Zeichen kürzen.

Zeile 1460: B$ löschen.

Zeile 1470: Falls A$ leer ist, sofort zurückspringen.

Zeile 1480: Das erste Zeichen von A$ in B$ speichern.

Zeile 1490: A$ um ein Zeichen kürzen.

Zeile 1500: Rücksprung.

Zeile 1510 - 1590: Hexadezimalzahl aus A$ in Z lesen.

Zeile 1510: Z zunächst löschen.

Zeile 1520: Ein Zeichen aus A$ lesen.

Zeile 1530 - 1540: Falls das Zeichen keine zulässige Hexziffer ist (0 - 9 oder A - F), Ende der Hexadezimalzahl.

Zeile 1550: Hexadezimalziffer in B$ umwandeln in Dezimalzahl in Z1.

Zeile 1560: Z entsprechend erhöhen.

Zeile 1570: Schleife wiederholen.

Zeile 1580: Falls Z größer als 32767 ist, muß es eine negative Zahl sein (damit PEEK und POKE funktioniert).

Zeile 1590: Rücksprung.

<u>Zeile 1600 - 1720:</u>	<u>Zahl in Z in hexadezimal umwandeln und in Z$ ablegen.</u>
Zeile 1600:	Falls Z größer als 32767 ist, muß es eine negative Zahl sein.
Zeile 1610:	Z1 = höherwertiges Byte von Z.
Zeile 1620:	Z2 = niederwertiges Byte von Z.
Zeile 1630:	Z$ als leer initialisieren.
Zeile 1640 - 1650:	Höchstwertige Ziffer erzeugen.
Zeile 1660 - 1670:	Zweite Ziffer erzeugen.
Zeile 1680 - 1690:	Dritte Ziffer erzeugen.
Zeile 1700:	Letzte Ziffer erzeugen.
Zeile 1710:	Zahl in Z3 in hexadezimale Ziffer umwandeln und an Z$ anhängen.
Zeile 1720:	Rücksprung.

<u>Zeile 1730 - 2080:</u>	<u>Write SYSTEM-Tape</u>
Zeile 1730 - 1780:	Maschinenprogramm in WR$ einlesen.
Zeile 1790:	X zeigt auf Eintrag der Variable WR$. In X steht die Länge des Strings. In X+1 steht das niederwertige Byte der Adresse, bei der der String im Speicher beginnt. In X+2 steht das höherwertige Byte der Adresse, bei der der String im Speicher beginnt.
Zeile 1800:	X = Adresse, bei der der Stringtext im Speicher beginnt.
Zeile 1810:	Adresse in X für PEEK und POKE anpassen.
Zeile 1820:	R = Adresse eines Unterprogramms innerhalb des Maschinenprogramms. (Genauere Erläuterungen zu diesem Programm siehe nächstes Kapitel)
Zeile 1830:	R1 = höherwertiges Byte von R.
Zeile 1840:	R2 = niederwertiges Byte von R.
Zeile 1850 - 1880:	Unterprogrammadresse in Programm POKEn.
Zeile 1890 - 1930:	Start-, End- und Einsprungadresse einlesen und in die ersten sechs Bytes des Maschinenprogramms POKEn.
Zeile 1940 - 1950:	Programmnamen in PN$ einlesen.
Zeile 1960 - 1980:	Byte Nr. 6 bis Byte Nr. 11 des Maschinenprogramms mit Leerzeichen füllen.
Zeile 1990 - 2010:	Den Programmnamen in die Bytes 6 bis maximal 11 des Maschinenprogramms POKEn.
Zeile 2020 - 2060:	Startadresse des Programms für USR-Aufruf POKEn.
Zeile 2070:	Maschinenprogramm aufrufen.
Zeile 2080:	Zur Kommandoeingabe springen.

Zeile 2090 - 2130: Das Maschinenprogramm für Write Tape.

Zeile 2090 - 2120: Das eigentliche Programm.
Zeile 2130: Das Unterprogramm zum Schreiben eines Datenblocks.

Zeile 2140 - 2200: Eine Zahl aus A$ lesen und in X und X+1 POKEn.

Zeile 2140: Eine Zahl aus A$ in Z lesen.
Zeile 2150: Z1 = höherwertiges Byte von Z.
Zeile 2160: Z2 = niederwertiges Byte von Z.
Zeile 2170: Das niederwertige Byte wird in X gePOKEd.
Zeile 2180: Das höherwertige Byte wird in X+1 gePOKEd.
Zeile 2190: X wird um 2 erhöht.
Zeile 2200: Rücksprung.

Zeile 2210 - 2240: Alle führenden Leerzeichen aus A$ löschen.

Zeile 2210: Ein Zeichen aus A$ in B$ lesen.
Zeile 2220: Wenn BS ein Leerzeichen enthält, Prozedur wiederholen.
Zeile 2230: A$ wieder korrigieren.
Zeile 2240: Rücksprung.

Nach der Erläuterung des Basic-Teils des Monitorprogramms hier nun noch ein Listing und die Erläuterungen zu dem Maschinenprogramm, das die Bandaufzeichnung macht.

```
8404 2A 8E 40     LD   HL,(408EH)    *.@
8407 11 FA FF     LD   DE,FFFAH      ...
840A 19           ADD  HL,DE         .
840B CD 3F 02     CALL 023FH         .?.
840E 3E 55        LD   A,55H         >U
8410 CD 1F 02     CALL 021FH         ...
8413 06 06        LD   B,06H         ..
8415 7E           LD   A,(HL)        .
8416 23           INC  HL            #
8417 CD 1F 02     CALL 021FH         ...
841A 10 F9        DJNZ 8415H         ..
841C 11 F4 FF     LD   DE,FFF4H      ...
841F 19           ADD  HL,DE         .
8420 E5           PUSH HL            .
8421 DD E1        POP  IX            ..
8423 DD 5E 00     LD   E,(IX+00H)    .^.
8426 DD 56 01     LD   D,(IX+01H)    .V.
8429 DD 6E 02     LD   L,(IX+02H)    .n.
842C DD 66 03     LD   H,(IX+03H)    .f.
842F AF           XOR  A             .
8430 ED 52        SBC  HL,DE         .R
8432 23           INC  HL            #
8433 7C           LD   A,H           .
8434 B7           OR   A             .
8435 28 08        JR   Z,843FH       (.
8437 06 00        LD   B,00H         ..
8439 CD 57 84     CALL 8457H         .W.
843C 25           DEC  H             %
843D 18 F4        JR   8433H         ..
843F 7D           LD   A,L           .
8440 45           LD   B,L           E
8441 B7           OR   A             .
8442 C4 57 84     CALL NZ,8457H      .W.
8445 3E 78        LD   A,78H         >x
8447 CD 1F 02     CALL 021FH         ...
844A DD 7E 04     LD   A,(IX+04H)    ...
844D CD 1F 02     CALL 021FH         ...
8450 DD 7E 05     LD   A,(IX+05H)    ...
8453 CD 1F 02     CALL 021FH         ...
8456 C9           RET                .
8457 3E 3C        LD   A,3CH         ><
8459 CD 1F 02     CALL 021FH         ...
845C 78           LD   A,B           x
845D CD 1F 02     CALL 021FH         ...
8460 7B           LD   A,E           {
8461 CD 1F 02     CALL 021FH         ...
8464 4F           LD   C,A           O
8465 7A           LD   A,D           z
8466 CD 1F 02     CALL 021FH         ...
8469 81           ADD  A,C           .
846A 4F           LD   C,A           O
846B 1A           LD   A,(DE)        .
846C 13           INC  DE            .
846D CD 1F 02     CALL 021FH         ...
8470 81           ADD  A,C           .
8471 4F           LD   C,A           O
8472 10 F7        DJNZ 846BH         ..
8474 C3 1F 02     JP   021FH         ...
```

In dem Listing auf der vorherigen Seite beginnt das Programm bei Adresse 8404H. Dies kann jedoch nicht immer so sein, da die Strings immer an anderen Stellen abgelegt werden. Dies hängt zum Beispiel davon ab, ob Ihr Rechner über 16 oder 32 K RAM verfügt, ob obere RAM-Bereich durch Eingabe einer MEM SIZE geschützt wurden etc. Hierdurch taucht ein großes Problem auf: Das Maschinenprogramm muß zum einen auf Adressen zugreifen, in die das Basic-Programm die Start-, End- und Einsprungadresse des auf Band zu speichernden Speicherbereichs sowie den Programmnamen abgelegt hat. Zum anderen wird in dem Programm zweimal ein Unterprogramm aufgerufen. Auch diese Adresse muß korrigiert werden. Dieses Problem löst schon der Basic-Teil der Write SYSTEM Tape Routine. In den Zeilen 1820 bis 1880 wird aus der Anfangsadresse des Programms der Anfang des Unterprogramms berechnet und an den entsprechenden Stellen im eigentlichen Programm abgelegt (in diesem Listing wären dies die Adressen 843AH und 8443H). Um die Stellen zu finden, an denen das Basic-Programm die nötigen Adressen und den Programmnamen abgelegt hat, wird derselbe Trick benutzt, diesmal jedoch vom Maschinenprogramm aus.
In Adresse 8404 wird das HL-Register mit dem Inhalt der Speicherzellen 408EH/408FH geladen. Diese Speicherzellen enthalten die Startadresse der Routine, die mit dem USR-Befehl aufgerufen werden soll (kennen Sie sicher: 16526/16527 dez.). In diesem Fall würde HL also 8404H enthalten, denn dies ist ja die Startadresse des Programms!
Die nächsten zwei Befehl erniedrigen HL um 6, sodaß es nun auf den Beginn des Programmnamens im Speicher zeigt. Nun wird zunächst eine ROM-Routine aufgerufen, die einen sogenannten Leader auf das Band schreibt. Dieser Leader sorgt dafür, daß beim Lesen alle Bytes genauso gefunden werden, wie sie geschrieben wurden (siehe nächstes Kapitel: Format von SYSTEM- und CLOAD-Bändern). Die Befehlsfolge LD A,55H; CALL 021FH zeichnet das Byte 55H auf Kassette auf. Die Befehle in den Adressen 8413H bis 841BH stellen eine Schleife dar, die den Programmnamen auf Kassette aufzeichnet. Die Befehle in Adresse 841CH bis 842EH laden die Startadresse des Programms in DE und die Endadresse in HL. Nun wird die Differenz berechnet und in HL abgelegt. Adresse 8433H bis 843EH stellt wieder eine Schleife dar, in der soviele 256 Byte lange Blöcke vom Speicher auf Kassette übertragen werden, wie durch die Programmlänge angegeben wurden. Dann wird noch (843FH bis 8444H), falls nötig, ein kürzerer Block ausgegeben, der die restlichen Bytes enthält. Zum Abschluß wird die Einsprungadresse aufgezeichnet und in den Basic-Teil des Monitorprogramms zurückgesprungen.
Das Unterprogramm, das einen Block aufzeichnet, hat folgende Form:
Zunächst wird ein Byte 3CH aufgezeichnet, dann die Blocklänge aus dem B-Register. Dann folgen das niederwertige und höherwertige Byte der Startadresse des Blocks, dann alle Datenbytes. Zum Schluß wird noch die Prüfsumme ausgegeben (Siehe nächstes Kapitel).

Das Format von SYSTEM-Bändern:

Ein Band, das mit dem SYSTEM-Befehl eingelesen werden soll, muß folgendes Format haben:

1. Leader — Dieser Leader besteht aus 256 Bytes AAH und einem Byte 66H. Der Sinn dieses Leaders ist folgender:
Jedes Byte wird bitweise auf Kassette aufgezeichnet. Es ist also wichtig, das der Rechner an der richtigen Stelle anfängt, zu lesen, um die richtigen Bits zu einem Byte zusammenzufassen. Dabei hilft der Leader. Der Rechner liest immer ein Bit vom Band und schiebt es durch das A-Register. Solange die Bytes AAH gelesen werden, enthält das A-Register entweder AAH oder 55H, aber nie 66H. In dem Moment, in dem das A-Register 66H enthält, weiß der Rechner, daß er die Bits nun richtig in Gruppen zusammenfassen kann.

2. Ein Byte 55H: Dieses Byte dient als Kennung für SYSTEM-Bänder.

3. Der Programmname — Die nächsten 6 Bytes geben den Programmnamen an. Falls dieser kürzer als 6 Zeichen ist, so sind die letzten Zeichen Leerzeichen.

4. Blockkennung: Als nächstes Byte folgt die sogenannte Blockkennung. Ein Byte 3CH gibt an, daß nun ein Datenblock folgt, ein Byte 78H gibt an, daß nun die Einsprungadresse des Programms folgt. Alle anderen Bytes werden ignoriert und es wird weiter nach 3CH oder 78H gesucht.

5. Datenblocks: Das erste Byte in einem Block gibt die Anzahl von Datenbytes an, die dieser Block enthält. 00H entspricht dabei 256 dez.
Das zweite und dritte Byte bilden die Adresse, ab der die Datenbytes im Speicher abgelegt werden sollen.
Dann folgen die Datenbytes und zum Schluß noch ein Byte, die Checksum.
Diese Checksum ist das niederwertige Byte der Summe aller Datenbytes plus das höher- und niederwertige Byte der Ladeadresse.

6. Einsprung: Nach dem Byte 78H folgen das niederwertige und das höherwertige Byte der Einsprungadresse des Programms.

Das Format von CLOAD-Bändern.

Bändern, die mit dem CLOAD-Befehl geladen werden sollen, müssen folgendes Format haben:

1. Leader

2. Drei Bytes D3H

3. 1 Byte für den Programmnamen

4. Der Programmtext, wie er im Speicher steht. Ende, wenn Link 0000H erreicht.

Wichtig ist, daß der CLOAD-Befehl kein Prüfsummen akzeptiert und also das Gelesene in keiner Weise auf Richtigkeit überprüft.

Die Begriffe "LSB" und "MSB"

Wir gehen davon aus, daß Ihnen die Begriffe Bit, Byte und Hexadezimalsystem vertraut sind. Sollte dies jedoch nicht der Fall sein, lesen Sie vorab Kapitel 31 des Handbuches "COLOUR BASIC - leicht gelernt".

Was bedeuten nun die Abkürzungen LSB und MSB ?

LSB ist die englische Abkürzung für "Lower Significant Byte", was soviel bedeutet wie niederwertiges Byte.
Entsprechend ist MSB die Abkürzung von "More Significant Byte", und bedeutet höherwertiges Byte.

LSB und MSB bilden zusammen eine 16-Bit Adresse mit der alle Speicheradressen angesprochen werden können.
64 K-Byte sind genau 65536 Byte die mit Adressen zwischen 0000H und FFFFH (0 bis 65535) angesprochen werden.
Wenn man z.B. Adresse 44E2H ansprechen will, so ist das MSB=44H und das LSB=E2H.
Die dezimale Schreibweise ergibt sich aus:
44H = 68 dez. und E2H = 226 dez., also 68*256+226 = 17634.
Man multipliziert also das MSB mit 256 und addiert das LSB hinzu.

In der Z-80 Maschinensprache werden die beiden Adreßbytes in der Reihenfolge LSB/MSB abgearbeitet. Also z.B. JP 0066H muß als C3 66 00 eingegeben werden.
Der Basic-Interpreter verhält sich beim Gebrauch von Adressen und Variablen genauso (siehe auch "Links" und "Zeilennummern" im folgenden Kapitel).

Wie werden Basicprogramme abgespeichert ?

Sie kennen natürlich den Basic-Befehl "LIST". Dieser Befehl zeigt Ihnen in klar lesbarer Form das Basicprogramm an, das im Speicher steht.
Wie aber "merkt" sich der Computer das Programm intern - so wie Sie es auf dem Bildschirm sehen oder etwa anders?
Und woher weiß das Colour-Genie, wo es das Programm abspeichern soll?
Diesen Fragenkomplex wollen wir nun klären. Am Ende finden Sie dann eine interessante Anwendung.

1. Die Basicprogramm-Anfangsadresse:
   In den beiden Speicherzellen 40A4H und 40A5H (16548 und 16549 dez.) steht die Adresse, bei der das Basicprogramm im Speicher beginnt.

   Haben Sie beim Einschalten nicht <MOD SEL> niedergehalten, beginnt das Basicprogramm bei 5801H (22529 dez.).
   Dies läßt sich einfach überprüfen:

```
READY
>PRINT PEEK(&H40A4)+256*PEEK(&H40A5)
 22529
READY
>PRINT &H5801
 22529
READY
>
```

   Wenn Sie beim Einschalten die <MOD SEL>-Taste gedrückt haben, d.h. wenn der Speicher von 4800H (18432 dez.) bis 57FFH (22527 dez.) nicht für die hochauflösende Grafik verwendet werden soll, beginnt das Basicprogramm bei Adresse 4801H (18433 dez.).
   Auch dies läßt sich leicht überprüfen:

```
READY
>PRINT (PEEK(&H40A4)+256*PEEK(&H40A5))=&H4801
-1
```

   (Die -1 bedeutet, daß obige logische Aussage wahr ist.)

2. Die Speicherung eines Basicprogramms:
   Wir wissen nun, ab welcher Adresse ein Basicprogramm im Speicher beginnt.
   Die Frage ist nun: Wie wird das Programm dort gespeichert.
   Um dies zu untersuchen, laden Sie den "Basic-Monitor", den weiter vorne aufgelistet finden, und den Sie auf ein Band gespeichert haben sollten. Wenn Sie dies noch nicht getan haben, holen Sie dies jetzt nach.
   Wir wollen nun mit dem "Basic-Monitor" untersuchen, wie der Monitor selbst abgespeichert wird.

Wenn Sie beim Einschalten nicht <MOD SEL> gedrückt hatten, beginnt das Programm im Speicher bei Adresse 5801H. Starten Sie nun den "Basic-Monitor" mit "RUN". Geben Sie dann "H5801" ein - der Monitor macht nun ein hexadezimales Listing. Brechen Sie dieses nach der 7. Zeile mit <CLEAR> ab. Anschließend können Sie noch ein ASCII-Listing dieses Speicherbereiches machen. Dazu geben Sie das Kommando "A5801" ein. Dieses Listing können Sie nach der 3. Zeile ebenfalls mit <CLEAR> abbrechen. Ihr Bildschirm sieht dann etwa so aus (Die Linien sind nachträglich zur Orientierung eingezeichnet; darunter sind hier noch die ersten vier Zeilen des "Basic-Monitors" zum Vergleich aufgelistet) :

```
Command? H5801
5801 07 58 0A 00 84 00 12 58    57
5809 14 00 B8 20 31 30 30 20    AD
5811 00 22 58 1E 00 B2 22 43    AF
5819 6F 6D 6D 61 6E 64 22 3B    D9
5821 00 2B 58 28 00 89 20 41    95
5829 24 00 33 59 72 00 8F 41    B7
5831 24 05 22 22 CA 33 30 00    6A
Command? A5801
5801  ::     :.     1000 ":'    "C
5819 ommand'  +::'    A$ 9::2   H
5831 $█ ""█30 P.:   E90  .:F   B$
Command?
```

```
10 CLS
20 CLEAR 1000
30 PRINT"Command";
40 INPUT A$
```

Jede Basiczeile hat im Speicher nun folgendes Format:

1. 2 Bytes, die die Anfangsadresse der nächsten Zeile angeben. ("Link")
2. 2 Bytes, die die Zeilennummer in hexadezimal enthalten.
3. Der Text der Zeile, wobei alle Befehle in sogenannte "Tokens" verschlusselt werden. D.h., daß ein Befehl nicht als Zeichenfolge gespeichert wird, sondern als ein oder zwei Bytes. Eine Liste dieser Tokens mit den zugehörigen Befehlen finden Sie in Anhang A.
   Alles, was kein Befehl ist, wird in normaler Textform abgespeichert.
4. Ein Byte 00H als Zeilen-End-Kennung.

Das Programmende ist durch einen Link markiert, der 00 00 ist.
Dies klingt alles recht kompliziert, aber anhand des oben ausgedruckten Beispiels wollen wir Ihnen das Prinzip nun verdeutlichen.

Sehen wir uns zuerst einmal die erste Basic Zeile an:
10 CLS
Im Speicher steht:

5801:	07 58	Dies ist der Link, d.h. die nächste Zeile beginnt bei Adresse 5807H. Beachten Sie, daß die Adresse 5807H, wie üblich, als 07 58 gespeichert (siehe voriges Kapitel LSB/MSB).
5803:	0A 00	Dies ist die Zeilennummer: 000AH = 10 dez.
5805:	84	Dies ist das Token für den Befehl CLS, wie Sie anhand Anhang A und Anhang B leicht überprüfen können. (84H = 132 dez.)
5806:	00	Ende der Zeile

Genauso läßt sich die zweite Programmzeile untersuchen:
20 CLEAR 1000
Im Speicher steht:

5807:	12 58	Die nächste Zeile beginnt bei Adresse 5812H.
5809:	14 00	Zeilennummer: 0014H = 20 dez.
580B:	B8	Token fur CLEAR (B8H = 184 dez.).
580C:	20 31 30 30 30	ASCII-Text ' 1000'.
5811:	00	Zeilenende.

Analog dazu können Sie auch die zwei nächsten Zeilen erkennen, sowie naturlich jede andere Basic-Zeile auch.

Wie Sie Anhang A entnehmen können, sind die Dezimalwerte aller Tokens größer als 127, d.h. ein Token ist durch ein gesetztes Bit Nummer 7 gekennzeichnet (Im Gegensatz zu den ASCII-Zeichen, die alle einen Wert kleiner als 128 haben - vergleiche dazu Anhang B).
Daraus ergibt sich, daß mit einem Byte nur maximal 128 Befehle codiert werden können, da das 7. Bit ja immer gesetzt sein muß. Beim großen Befehlssatz des Colour-Genies reichte dies jedoch nicht aus. Aus diesem Grunde wurde für eine Reihe von Befehlen eine Codierung mit zwei Bytes gewählt, wobei das erste Byte immer FFH (255 dez.) ist.
Der Befehl PLOT hat z.B. das Doppeltoken FFH 84H (255 dez. 132 dez.).
In diesem Zusammenhang seien noch drei Besonderheiten der Abspeicherung erwähnt:

1. Beim Befehl ELSE wird intern vor dem ELSE noch ein Doppelpunkt abgespeichert, der beim LISTen aber nicht ausgegeben wird.
2. Der Befehl FCOLOUR wird durch 3 Bytes ausgedrückt: FFH 81H 52H (255 dez. 129 dez. 82 dez.)
3. Die Kurzform des REM-Befehls (') wird auf folgende Weise abgespeichert:
   3AH 93H FBH (58 dez. 147 dez. 251 dez.)

Als letztes wollen wir nun eine interessante Anwendung für das in diesem Kapitel Erklärte bringen.
Es kommt öfters vor, daß man in einem Basicprogramm Teile des Programms selbst ändern möchte. Dies ist nicht ohne weiteres möglich - normalerweise muß man die entsprechende Zeile erst eingeben bzw. editieren und muß dann das Programm neu starten.
Wenn man allerdings weiß, wie Basicprogramme abgespeichert werden, kann man das Programm von sich aus ändern.
Ein wesentlicher Vorteil neben dem Eingabe-Komfort ist, daß bei dieser Methode keine Variablenwerte gelöscht werden.
Folgendes Programm ist nun ein Beispiel für dieses Verfahren.
Das Programm fragt nach einer mathematischen Funktion. Diese geben Sie ganz normal ein und sie wird dann in das Programm "gePOKEd". Anschließend werden noch Unter- u. Obergrenze, sowie die Schrittweite abgefragt. Das Programm berechnet dann eine Wertetabelle für die Funktion.
Natürlich gibt es eine Menge anderer Anwendungen - lassen Sie mal Ihre Phantasie spielen !
Wichtig kann die Kenntnis des Abspeicherverfahrens auch sein, wenn man ein kaputtes Basicprogramm, in dem z.B. falsche Zeilennummern vorkommen, reparieren will.

Programmlisting:

```
10 CLEAR500:DEFINTA-V:CLS:GOTO30
20 Y=SIN(X):RETURN:REM::::::::::::::::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::
30 INPUT"Funktion: Y = ";F$
40 F$="Y="+F$
50 T$=""
60 PRINT@40,F$;CHR$(30):X=6040:T=205
70 IFX=6095THEN130ELSER$=CHR$(PEEK(X)AND127):X=X+1
80 IFPEEK(X)>127THEN100
90 R$=R$+CHR$(PEEK(X)):X=X+1:GOTO80
100 IFLEN(F$)<LEN(R$)THENT=T+1:GOTO70
110 IFLEFT$(F$,LEN(R$))<>R$THENT=T+1:GOTO70
120 T$=T$+CHR$(T):F$=RIGHT$(F$,LEN(F$)-LEN(R$)):IFF$<>""THEN60ELSE140
130 T$=T$+LEFT$(F$,1):F$=RIGHT$(F$,LEN(F$)-1):IFF$<>""THEN60
140 PRINT@40,;:T$=T$+":"+CHR$(146)+":"+CHR$(147)
150 X=PEEK(&H40A4)+256*PEEK(&H40A5)
160 X1=PEEK(X)+256*PEEK(X+1)
170 IFPEEK(X1+2)+256*PEEK(X1+3)<>20THENX=X1:GOTO160
180 X1=X1+4
190 FORX=1TOLEN(T$):POKEX1,ASC(MID$(T$,X,1)):X1=X1+1:NEXTX
200 ONERRORGOTO210:GOSUB20:ONERRORGOTO0:GOTO230
210 IFERR=5ORERR=11THEN230ELSECOLOUR3:PRINT"Fehlerhafte Funktion":CALL
357C:COLOUR2
220 IFINKEY$=""THEN220ELSERUN
230 INPUT"Untergrenze";XU
240 INPUT"Obergrenze";XO
250 IFXU>=XOTHEN230
260 INPUT"Schrittweite";XS
270 IFXS<=0THEN230
280 FORX=XUTOXOSTEPXS
290 GOSUB20:PRINT"F(";X;") =";Y
300 IFINKEY$=""THEN300ELSENEXTX
310 IFINKEY$=""THEN310ELSERUN
```

Erklärung des Programms:

Zeile 10 : Genug Platz für Strings schaffen, zur Beschleunigung des Programmablaufs alle nicht benutzten Variablen als Integer erklären, den Bildschirm löschen und das eigentliche Programm starten.

Zeile 20 : In dieser Zeile soll die Funktion als Unterprogramm abgelegt werden. Die ':' halten den Platz für längere Funktionen frei. Das Unterprogramm besteht immer aus der Berechnung eines Wertes Y aus einer Funktion von X. Das REM verhindert, daß etwaige Reste längerer Funktionen zu einem SYNTAX Error führen.

Zeile 30 : Die Funktion wird in F$ eingelesen.

Zeile 40 : Der erste Teil des Unterprogramms ist Y=f(X).

Zeile 50 : In T$ wird die codierte Zeile generiert.

Zeile 60 : Den noch zu codierenden Rest der Funktion ausgeben. X zeigt auf den Anfang einer Tabelle im ROM, in der sämtliche BASIC-Schlüsselworte stehen (diese Tabelle beginnt bei 1650H, in diesem Fall wurde jedoch erst bei den arithmetischen Schlüsselworten begonnen). T entspricht dem Token des momentanen Schlüsselwortes.

Zeile 70 : Wenn X = 6095, dann ist das Ende der arithmetischen Schlüsselworte erreicht, also wurde kein passendes gefunden. Weiter Verarbeitung in Zeile 130.

Zeile 70 (Rest) - Zeile 90 :
Das momentane Schlüsselwort wird in R$ eingelesen. Der erste Buchstabe des nächsten Schlüsselwortes ist durch ein gesetztes Bit 7 gekennzeichnet, wenn ein solches Zeichen gefunden wird, enthält R$ das komplette Schlüsselwort.

Zeile 100: Wenn der noch zu codierende Rest der Funktion kürzer als das gefundene Schlüsselwort ist, kann es nicht passen. Nächstes Schlüsselwort versuchen!

Zeile 110: Falls der Rest der Funktion das Schlüsselwort nicht enthält, nächstes Wort versuchen!

Zeile 120: Passendes Wort gefunden. Token an T$ anhängen, F$ entsprechend kürzen. Wenn FS noch nicht leer ist, weitermachen, sonst ist die Codierung beendet.

Zeile 130: Diese Zeile wird angesprungen, wenn kein Token passt. Ein Zeichen aus F$ wird in T$ übertragen, F$ wird gekürzt. Wenn F$ noch nicht leer ist, weitermachen, sonst ist die Codierung beendet.

Zeile 140: An T$ wird noch der Code für ':RETURN:REM' angehängt, um die Basic-Zeile zu komplettieren.

Zeile 150: X zeigt auf die erste Programmzeile.

Zeile 160 - Zeile 170:
Zeile 20 wird im Programm gesucht. Die Anfangsadresse steht dann in X1.

Zeile 180: Link und Zeilennummer werden übersprungen.

Zeile 190: T$ wird in Zeile 20 kopiert.
Zeile 200 - Zeile 220 :
Zeile 20 wird auf Fehler getestet. Die Zeile wird ausgeführt (GOSUB 20). Wenn kein Fehler auftaucht, wird zu Zeile 230 gesprungen, vorher jedoch noch die Fehlerabfrage abgestellt (ON ERROR GOTO 0:GOTO 230). Wenn ein Fehler auftaucht, so darf dies nur der FC-Error (Fehlercode 5) oder der /0-Error (Fehlercode 11) sein, sonst ist die Funktion fehlerhaft.

Zeile 230: In XU wird die Untergrenze eingelesen.
Zeile 240: In XO wird die Obergrenze eingelesen.
Zeile 250: Falls die Untergrenze größer oder gleich der Obergrenze ist, wird die Eingabe wiederholt.
Zeile 260: In XS wird die Schrittweite eingelesen.
Zeile 270: Falls die Schrittweite kleiner oder gleich Null ist, wird die Eingabe wiederholt.
Zeile 280: Schleife von XU bis XO mit der Schrittweite XS.
Zeile 290: Aus X einen Y Wert berechnen und ausgeben.
Zeile 300: Auf einen Tastendruck warten, dann nächsten Wert berechnen und ausgeben.
Zeile 310: Nach Beendigung der Schleife auf Tastendruck warten, dann Programm neu starten.

Wie werden Basic-Variablen abgespeichert?

Die Basic-Variablen werden in zwei Untergruppen aufgeteilt, nämlich:
- die normalen Variablen (z.B. A = 0) und die
- Feldvariablen (z.B. DIM A(100))

Die normalen Variablen werden direkt hinter dem Basic-Programm abgelegt (dies ist auch der Grund, weshalb nach jeder Veränderung des Programms alle Variablen gelöscht sind, denn das Programm kann ja dadurch seine Länge verändern). Die Anfangsadresse dieses Speicherbereichs (und damit natürlich auch die Endadresse Ihres Basicprogramms) steht in den Speicherzellen 40F9H/40FAH (16633/16634 dez.). In diesem Speicherbereich stehen die einzelnen Variablen nun einfach hintereinander, und zwar in folgendem Format:

1 Byte Typcode	Dieses Byte ist so gewählt, daß es auch die Länge des Datenfeldes der Variable angibt: 02 = Integer 03 = String 04 = einfache Genauigkeit 08 = doppelte Genauigkeit
2 Bytes Name	Der Name wird umgekehrt abgespeichert, das heißt, die Variable A1 heißt im Speicher 1A. Einbuchstabige Variablen wie A, B u.ä. werden mit einem Byte 00H an erster Stelle abgespeichert (z.B. 00H 41H = Variable A). Der Sinn dieser Methode ist, daß in jedem Fall ein Kennbuchstabe direkt vor dem Datenfeld steht, dies erleichert den Vorgang des Suchens nach einer bestimmten Variable.
2 - 8 Bytes Datenfeld	Je nach Typ folgen nun die der Variable zugeordneten Daten: Integer Der Wert in LSB/MSB Format. Strings Das erste Byte gibt die Länge des Strings an, die beiden anderen, wo der String im Speicher beginnt.

Einfache Genauigkeit
Die ersten drei Bytes enthalten die Mantisse, das vierte Byte den Exponenten. Die Mantisse ist folgendermaßen strukturiert:

Das 3. Byte ist das höchstwertigste, dann folgt das zweite und das erste Byte ist das niederwertigste. Diese drei Bytes zusammen ergeben eine 24-bit Zahl, wobei das höchste Bit das Vorzeichen der Mantisse angibt (0 = positiv). Das höchste Bit des Exponenten gibt an, ob der Binärpunkt nach rechts oder nach links verschoben wird (0 = rechts), die restlichen sieben Bits geben an, um wieviele Stellen der Punkt verschoben wird.

Doppelte Genauigkeit
Die Struktur entspricht genau der oben geschilderten, doch die Mantisse ist insgesamt 56 Bits lang.

Nach der Variablentabelle folgt der Speicherbereich, in dem die Felder abgelegt werden. Der Beginn dieses Bereiches steht in den Adressen 40FBH/40FCH (16635/16636 dez.). Der Eintrag für ein Feld sieht nun folgendermaßen aus:

1 Byte Typcode	Wie bei Variablen: 02 = Integer 03 = String 04 = Einfache Genauigkeit 08 = Doppelte Genauigkeit
2 Bytes Name	Die Abspeicherung des Namens entspricht der bei normalen Variablen.
2 Bytes Länge	Die folgenden zwei Bytes geben an, wieviel Speicherplatz das Feld benötigt. Wenn man zu der Adresse, in der das erste dieser beiden Bytes steht, diese Länge + 2 addiert, erhält man die Adresse, bei der das nächste Feld im Speicher beginnt.
1 Byte Dimension	Dieses Byte gibt an, wieviele Dimension das Feld hat.

n * 2 Bytes Dimension	Nun folgt für jede Dimension deren Länge. Da jede Dimension von 0 bis zum maximalen Index angesprochen werden kann, steht hier der maximale Index + 1.
Datenfeld	Schließlich folgen die Daten, die das Feld enthält. Ihr Format entspricht dem der Daten von normalen Variablen. Die Daten sind folgendermaßen sortiert: Gehen wir von dem Feld A(2,1) aus. Dann steht im Speicher: A(0,0); A(1,0); A(2,0); A(0,1); A(1,1); A(2,1)

Das Ende des Speicherbereichs, in dem die Felder abgelegt werden und somit der Beginn des freien Speichers steht in den Adressen 40FDH/40FEH (16637/16638 dez.)

Zusammenladen von mehreren Basicprogrammen

Sie kennen sicher das Problem, daß man zwei Basicprogramme aneinanderhängen möchte, ohne einen der beiden Teile neu eintippen zu müssen.
Abhilfe schafft folgendes Vorgehen:

1. Das Basicprogramm mit den kleinsten Zeilennummern mit CLOAD laden.
2. Eventuell Programm ändern...
3. Folgende Zeile eingeben (ohne Zeilennummer!):
   ```
   A=PEEK(&H40F9)+256*PEEK(&H40FA):A=A-2:POKE&H40A4,A-INT(A/256
   )*256:POKE&H40A5,INT(A/256)
   ```
   (Zur Erklärung der Adressen 40F9H/40FAH siehe Kapitel: "Wie werden Variablen abgespeichert?")
4. Nächstes Basicprogramm mit CLOAD laden. Wichtig ist, daß dieses Programm keine Zeilennummer enthält, die kleiner oder gleich der höchsten Zeilennummer ist, das als letztes geladen wurde.
5. Schritte 2 bis 4 sooft wiederholen, bis alle Teilprogramme einmal geladen wurden.
6. Folgende Zeile eingeben (ohne Zeilennummer!):
   ```
   POKE&H40A4,1:POKE&H40A5,88
   ```
7. Nun sind alle Programme vorhanden und Sie können mit Ihnen arbeiten.

Bitte beachten Sie:
Schritt Nr. 6 gilt nur, wenn Sie beim Einschalten nicht die MOD SEL-Taste gedrückt hatten und keine Programme wie den Colour-Compiler oder den Colour-Assembler geladen und gestartet haben. Dies alles verändert die Startadresse des Basicprogramms (siehe Kapitel "Wie werden Basicprogramme abgespeichert? "). In einem solchen Fall geben Sie vor Schritt 1 folgende Zeile ein (ohne Zeilennummer):
```
PRINTPEEK(&H40A4);PEEK(&H40A5)
```
und merken sich die beiden Zahlen, die nun ausgegeben werden. In Schritt Nr. 6 tauschen Sie nun die 1 gegen die erste und die 88 gegen die zweite der beiden Zahlen aus.

Reserviert

Sie hassen doch sicher auch die Situation, daß ein Programm Ihnen anzeigt: "Bitte geben Sie bei MEM SIZE? 32000 ein." Mit den folgenden Zeilen können Sie in Ihren Basicprogrammen eine Speicherobergrenze für das Basic neu festlegen. Sinnvoll ist dies, wenn man, kombiniert mit dem Basic, Maschinensprache-programme verwendet.

```
10 CLEAR 50
20 HA=PEEK(&H40B1)+256*PEEK(&H40B2)
30 HA=32000:'Je nach Bedarf festlegen
40 H2=INT(HA/256):H1=HA-H2*256
50 POKE&H40B1,H1:POKE&H40B2,H2
60 POKE&H40D6,H1:POKE&H40D7,H2
70 CLEAR50
80 REM Hier folgt Ihr Basicprogramm
```

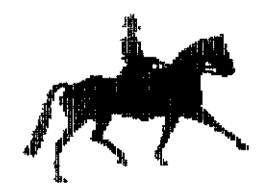
In Zeile 20 wird die alte Speicherobergrenze gelesen.
In Zeile 30 wird die neue Speicherobergrenze festgelegt.
(Hier 32000: Ihren Wünschen entsprechend ändern)
In Zeile 40 wird die 2-Byte-Integerzahl für die neue Obergrenze in zwei einzelne Bytes zerlegt.
In Zeile 50 und 60 werden diese neuen Werte in die Basicspeicheradresse und Stringvariablenadresse geschrieben.
Die Zeilen 10 und 70 sind notwendig und dürfen nicht entfernt werden.

Ausgabe der Bytes 0,11 oder 12 an den Drucker

Das Basic-ROM des Colour-Genie wandelt die Bytes 11 dez. (Top of form) und 12 dez. (Formfeed) in eine Folge von 10er (dez.) Bytes (Linefeeds) um.
Das Byte 0 wird überhaupt nicht ausgegeben.
(Zum Vergleich können Sie die entsprechende ROM-Routine mit einem Disassembler listen: Sie beginnt bei Adresse 04E7H und endet bei Adresse 0563H.)
Das heißt also, daß die Befehle LPRINT CHR$(0), LPRINT CHR$(11) und LPRINT CHR$(12) nicht die Werte 0,11 und 12 an den Drucker ausgeben!
Dies kann, je nach angeschlossenem Drucker, zu einigen Problemen führen. So kommt es bei der Programmierung von hochauflösenden Grafiken über Einzelnadelsteuerung (Bit-Image-Grafik) oft zu falschen Ausdrucken. Ferner ist die Tabulatorprogrammierung des Druckers oft beschränkt, da Byte 11 (dez.) das Steuerzeichen für einen vertikalen Tabulator ist. Eine anderes Problem ist z.B., daß man beim Star-Drucker DP 510 die Unterstreichung hilfs eines Bytes 00 abschaltet. Alle diese Probleme kann man mit folgenden Basic-Zeilen lösen, die mit GOSUB 100 aufgerufen werden. Der ASCII-Wert des auszugebenden Zeichens steht dabei in Variable X.

```
100 SOUND 7,127 : OUT248,15 : IF (INP(249) AND 239) = 47 THEN
    110 ELSE 100
110 SOUND 7,127 : SOUND 14,X : SOUND 7,255 : SOUND 15,0 : SOUND
    15,1 : RETURN
```

Zeile 100 prüft, ob der Drucker druckbereit ist. In Zeile 110 wird der Wert X an den Drucker ausgegeben. (Zur Ausgabe von Buchstaben vergleiche Anhang B.)
Natürlich können Sie die Zeilennummern so ändern, daß Sie in Ihr Programm passen. Als Anwendungbeispiel für obige Zeilen folgt nun ein kleines Programm, daß auf dem STAR-Drucker DP 510/DP 515 ein kleines Pferd ausdruckt:



Das Programm finden Sie auf der nächsten Seite aufgelistet. Beachten Sie, daß das Programm nur auf dem STAR-Drucker DP 510/515 läuft. In den Programmzeilen 260 und 270 finden Sie übrigens die oben aufgeführten Ausgabezeilen.

```
50 LPRINTCHR$(27);"8"
60 DEFINTA-Z
70 PRINT"Haben Sie etwas Geduld,ich drucke gleich"
80 FORZ=1TO10
90 LPRINTCHR$(27)"K"CHR$(86);:X=0:GOSUB260
100 FORC=1TO86:READX:GOSUB260:NEXT
110 LPRINTCHR$(27)"J"CHR$(16);:NEXT:LRPINT:END
120 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,7,7,15,15,15,3,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
130 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,64,65,195,255,255,255,255,243,96,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,1,1,1,3,3,3,3,7,15,16,0,0,0,0,0,0,0,0
140 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,249,255,255,255,255,255,255,255,15,1,1,1,1,1,0,0,0,0,0
,0,1,7,15,31,63,127,127,255,255,255,255,255,255,255,255,255,255,255,12
7,7,0,0,0,0,0,0,0
150 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,
1,0,0,0,0,0,1,7,31,255,255,255,255,255,255,255,255,255,255,207,207,255
,79,127,47,63,63,255,255,255,255,255,255,255,255,255,255,254,252,148,2
12,246,242,254,254,255,255,254,254,12
160 DATA0,0,0,0,0
170 DATA0,0,0,0,0,0,0,0,0,0,3,7,31,28,56,48,51,63,31,31,63,63,127,12
7,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,255
180 DATA255,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
190 DATA0,0,0,0,0,0,0,1,31,127,255,255,192,0,0,240,254,255,255,255,2
55,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,255,255,255,255,255
200 DATA255,255,255,255,247,3,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
210 DATA0,0,0,0,0,7,63,255,255,255,252,192,0,0,0,0,1,255,255,255,255
,254,252,248,240,192,192,225,255,255,255,254,248,224,0,0,0,0,128,128,1
28,128,192,192,192,224,224,224,192,192,128,128,128,135,255,255,255,252
,240,128,128,192,192,192,224
220 DATA224,240,240,248,248,252,124,31,15,7,3,1,0,0,0,0,0,0,0,0
230 DATA0,0,3,31,255,255,255,254,240,128,0,0,0,15,31,127,252,248,240
,192,0,0,0,0,0,0,32,240,240,248,248,28,30,15,7,3,1,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,3,63,254,248,224,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,128,19
2,224,240,112,62,31,7,7,3
240 DATA0,0,128,192,192,224,224,192,0,0,0,0,0,15,255,255,128,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,128,192,236,252,126,14,0,0,0,0,0,0,0,0,0,0,0,
240,254,254,14,6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0
250 DATA0,0,0,0,0,0,0,0,0,0,0,0,8,252,252,28,12,4,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
260 SOUND7,127:OUT248,15:IF(INP(249)AND239)=47THEN270ELSE260
270 SOUND7,127:SOUND14,X:SOUND7,255:SOUND15,0:SOUND15,1:RETURN
```

Ausgabe von Tabulatoren größer als 40 auf den Drucker

Der Basic-Interpreter Ihres Colour-Genie ist so aufgebaut, daß Sie an jeder Stelle einer Bildschirmzeile eine Tabulatormarke setzen können.
Analog dazu behandelt der Interpreter den Drucker.

Beachten Sie aber:

Eine Bildschirmzeile hat 40 Zeichen. Daher ist das Setzen einer Tabulatormarke auf eine Position > 39 mit dem TAB-Befehl auch auf dem Drucker nicht möglich. Benutzen Sie in solchen Fällen

statt:

```
10 LPRINT TAB(T);"Text"
```

folgendes:

```
10 LPRINT STRING$(T-PEEK(&H409B),32);"Text"
```

Wie beim TAB-Befehl gilt natürlich auch hier:
0<= T <= maximale Zeilenlänge - 1.
Beachten Sie, daß am Anfang eines Programms, in dem diese Methode benutzt wird, ein CLEAR Z stehen muß, wobei Z >= Zeilenlänge sein muß. Sonst könnte ein OS (Out of Stringspace) Error auftreten.

Interessante ROM-CALL's

Im folgenden sind einige nützliche ROM-Routinen aufgelistet und kurz erklärt, die dem Programmierer, der Erfahrung in Maschinenspracheprogrammierung hat, viel Arbeit abnehmen können.
Am besten schaut man sich die Routinen, die beschrieben sind, mit dem Disassembler eines Maschinensprache-Monitors an.
Im übrigen sei erwähnt, daß nur die wichtigsten Routinen aufgeführt sind, da in nächster Zeit ein komplettes, dokumentiertes ROM-Listing zur Verfügung stehen wird.

0000H System-Kaltstart

0008H Basic Syntax-Check. Wird über RST 08 aufgerufen
Das Byte hinter dem RST 08 Code wird mit (HL) verglichen. Wenn ungleich dann "?SN Error".

0018H CP HL,DE . Vergleicht das Doppelregister DE mit HL.
HL > DE C=0 Z=0
HL = DE C=0 Z=1
HL < DE C=1 Z=0

002BH INCH. Liest einen Tastendruck in Akku. A=0 keine Taste.

0033H OUTCH. Gibt den Akkuinhalt als Zeichen auf die aktuelle Bildschirmposition in 4020H/4021H aus. Zeichen kleiner 20H werden als Control-Zeichen akzeptiert (Cursor Home, Clear to End of Line usw.) und bei Bedarf wird der Bildschirminhalt gescrollt.

003BH PRTCHR. Gibt das Zeichen in A auf den Drucker aus.

0040H INLINE. Liest ab HL als Bufferadresse (norm. 41E8H) maximal B Zeichen von der Tastatur ein. Das letzte Zeichen ist 0DH (RETURN). Bei BREAK ist das Carryflag gesetzt. Bei RET steht HL wieder auf Buffer und B enthält die Zahl der eingegebenen Zeichen.

0049H Wie INCH, jedoch wird auf Tastendruck gewartet.

0060H Wartet BC-Registerinhalt * 14.66 µs.

0066H Basic-Warmstart. Legt den Basic-Stackpointer neu an und springt anschließend wieder zum Basic ">READY".

01C9H CLS. Löscht Bildschirm über 1CH = Cursor Home und 1FH = Clear to End of Frame Codes.

01D3H RANDOM. Erzeugt neue RND Werte.

01E4H	Läßt das rechte '*' blinken durch XOR 0AH .
01EDH	Read Byte. Ein Byte über Kassettenport in A lesen.
021FH	Write Byte. Akkuinhalt auf Kassettenport schreiben.
023FH	Write Sync. 256 * AAH und einmal 66H auf Kassettenport.
024CH	Read Sync. Wartet auf Synchronistation.
0314H	Liest ein Doppelbyte LSB/MSB von Kassette in HL.
0A7FH	CINT. Überträgt die Basicvariable X (4121H/4122H) in das HL Registerpaar.
0A9AH	Überträgt HL in die Variable X. (Anm.: X ist eine Variable, die vom Basic-Interpreter in den Adressen 4121H ff. angelegt wird und keine Variable des normalen Basic's !!!)
0FAFH	LINENR. Gibt HL als dez. Zahl ohne Vorzeichen aus (z.B. Zeilennummer).
1650H	ASCII Tabelle der Basic-Tokens. Erstes Byte jeweils mit gesetztem Bit 7. Sortiert nach aufsteigenden Tokens.
1822H	Sprungtabelle zu den Tokens. Enthält die Einsprungadressen der entsprechenden Routinen als Doppelbytes.
2B01H	Verwandelt Ausdruck ab (HL) in eine Integerzahl in DE. Nachher steht HL auf dem ersten Byte nach dem Ausdruck. Dient zur Übernahme von Zahlen oder Ausdrücken in Maschinensprache.
2B1BH	Verwandelt Ausdruck ab (HL) in eine Zahl < 256 dez. Wert steht hinterher im Akku, sonst wie 2B01H.
2B75H	Print String ab (HL). Bei (HL)=0 RET.
38A9H	FGR. Schaltet Bildschirm auf FGR um.
38B0H	LGR. Schaltet auf LGR um.
38CCH	COLOUR n. Wobei n-1 im Akku stehen muß (0...F).
38DAH	FCOLOUR n. Wobei n-1 im Akku stehen muß (0...3).
3FB0H	CHAR n. Schaltet auf CHAR n. Im Akku muß n-1 stehen.

Änderungen der Break-RST Tasten usw siehe Anhang

*** Neuer Zeichensatz ***

Hat Sie der eingebaute Zeichensatz Ihres Colour-Genie auch schon einmal gelangweilt? Dann geben Sie mit dem "Basic-Monitor" oder einem anderen Monitor das nachfolgende Hex-Listing ein (Die Pruefsummen am Ende jeder Zeile werden nicht mit eingegeben). Bevor Sie jedoch den Monitor laden, muessen Sie bei MEM SIZE den Wert 31936 eingeben, um das Programm zu schuetzen. Nachdem Sie das Programm eingegeben und alle Pruefsummen verglichen haben, schreiben Sie alles mit dem W-Befehl auf Kassette: W7CC0 7FFF 7CD2. Dann koennen Sie das Zeichensatzprogramm mit G7CD2 starten. Nun werden alle ASCII Zeichen in der neuen Schrift ausgegeben. Da die neue Schrift mit den definierbaren Zeichen erzeugt wird, duerfen Sie nur die Zeichen 128 bis 159 umdefinieren und muessen immer auf CHAR 1 bleiben. Den normalen Zeichensatz koennen Sie weiterhin darstellen, indem Sie den Wert des Zeichens in den Bildschirm POKEn.

Erklärung der Funktionsweise des Programms

Nach dem im folgenden abgedruckten Hex-Listing des Programmes, finden Sie ein disassembliertes Listing der Routine, die den neuen Zeichensatz initialisiert.
Bei 7CD2H werden HL,DE und BC für einen Transfer eines kleinen Programms von 7CC0H bis 7CD1H nach 43A0H vorbereitet. Dieses kleine Programm wird über den PRINT-DCB (Device-Control-Block) angesprungen, dessen Zeiger nach 43A0H 'verbogen' wird.
Wenn das Programm angesprungen wird, steht im C-Register des Z-80 der Wert des auszugebenden ASCII-Zeichens.
Wenn er kleiner als 20H oder größer als 7FH ist, bleibt er erhalten. Andernfalls wird 80H dazuaddiert, d.h. aus dem Zeichen '!' daß den ASCII-Code 21H hat wird also in A1H umgewandelt, während z.B. ein Cursor-Home-Code 1CH als 1CH erhalten bleibt, damit der Basic-Interpreter ihn noch erkennt.
Bei 7CDDH wird der PRINT-DCB nach Adresse 43A0H verbogen.
Der DCB ist eine Doppelbyte-Adresse, die vom Basic-Interpreter im RAM abgelegt wird. Normalerweise zeigt sie auf eine ROM-Adresse (hier 30E4H), kann aber bei bedarf geändert werden. Man kann also den Basic-Interpreter dazu veranlassen, bei einer Zeichenausgabe in ein eigenes Programm zu verzweigen.
Die Adressen der DCB's sind folgende:
401EH = Display 4016H = Keyboard 4026H = Printer
Anschließend werden bei Adresse 7CE3H HL,DE und BC für den Transfer des Zeichensatzes von 7D00H bis 7FFFH nach F500H vorbelegt. Nach dem Transfer mit LDIR wird noch der Zeichensatz (CHARn) über die ROM-Routine bei 3FB0H geändert. Im A-Register steht dabei der CHAR-Wert minus eins. Für CHAR 1 also 0, was hier über XOR A erreicht wird. Zuletzt folgt ein Sprung zum BASIC-Warmstart bei Adresse 0066H. Von nun an werden alle Zeichen im Bereich zwischen 20H und 7FH auf A0H bis FFH erhöht. Wenn Sie wollen können Sie die Zeichen natürlich auch noch abändern, z.B. mit Hilfe eines Zeicheneditors.

```
7CC0 F5 79 FE 20 38 07 FE 80   49
7CC8 30 03 C6 80 4F F1 C3 E4   60
7CD0 30 00 21 C0 7C 11 A0 43   81
7CD8 01 11 00 ED B0 21 A0 43   B3
7CE0 22 1E 40 21 00 7D 11 00   2F
7CE8 F5 01 00 03 ED B0 AF CD   12
7CF0 B0 3F C3 66 00 00 00 00   18
7CF8 00 00 00 00 00 00 00 00   00
7D00 00 00 00 00 00 00 00 00   00
7D08 18 18 18 18 18 00 18 00   90
7D10 24 24 24 00 00 00 00 00   6C
7D18 24 24 7E 24 7E 24 24 00   B0
7D20 18 7E 40 7E 02 7E 18 00   EC
7D28 60 66 0C 18 30 66 06 00   86
7D30 30 48 48 30 6A 64 3A 00   F8
7D38 18 18 18 00 00 00 00 00   48
7D40 0C 18 30 30 30 18 0C 00   D8
7D48 30 18 0C 0C 0C 18 30 00   B4
7D50 18 5A 3C 18 3C 5A 18 00   74
7D58 00 18 18 7E 18 18 00 00   DE
7D60 00 00 00 00 18 18 30 00   60
7D68 00 00 00 7E 00 00 00 00   7E
7D70 00 00 00 00 00 18 18 00   30
7D78 00 04 08 10 30 60 00 00   AC
7D80 7C 44 4C 54 62 62 7E 00   A2
7D88 10 10 10 10 18 18 18 00   88
7D90 7C 04 04 7C 60 60 7E 00   3E
7D98 7C 04 04 1E 06 06 7E 00   2C
7DA0 42 42 42 7E 06 06 06 00   56
7DA8 7E 40 40 7E 06 06 7E 00   06
7DB0 40 40 40 7E 62 62 7E 00   80
7DB8 7E 02 02 02 06 06 06 00   96
7DC0 7C 44 44 3C 62 62 7E 00   82
7DC8 7E 42 42 7E 06 06 06 00   92
7DD0 00 00 18 00 18 00 00 00   30
7DD8 00 00 18 00 18 18 30 00   78
7DE0 0C 18 30 60 30 18 0C 00   08
7DE8 00 00 7E 00 7E 00 00 00   FC
7DF0 60 30 18 0C 18 30 60 00   5C
7DF8 3C 66 06 0C 18 00 18 00   E4
7E00 7E 42 5E 4C 60 62 7E 00   AA
7E08 7E 42 42 7E 62 62 62 00   A6
7E10 7E 42 42 7C 62 62 7E 00   C0
7E18 7E 42 40 40 60 62 7E 00   80
7E20 7C 46 42 42 62 66 7C 00   8A
7E28 7E 40 40 78 60 60 7E 00   B4
7E30 7E 40 40 78 60 60 60 00   96
7E38 7E 42 40 4E 62 62 7E 00   90
7E40 42 42 42 7E 62 62 62 00   6A
7E48 10 10 10 10 18 18 18 00   88
7E50 7E 42 02 02 06 46 7E 00   8E
7E58 42 44 48 78 64 62 62 00   6E
7E60 40 40 40 40 60 60 7E 00   3E
7E68 76 4A 4A 42 62 62 62 00   72
7E70 72 4A 4A 46 62 62 62 00   72
7E78 7E 42 42 42 62 62 7E 00   86
7E80 7E 42 42 7E 60 60 60 00   A0
7E88 7E 42 42 42 6A 64 7A 00   8C
7E90 7E 42 42 7E 68 64 62 00   AE
7E98 7E 42 40 7E 06 46 7E 00   48
7EA0 7E 52 10 10 18 18 18 00   38
7EA8 42 42 42 42 62 62 7E 00   4A
```

7EB0	42	42	42	42	64	68	30	00	04
7EB8	42	42	42	42	6A	6A	34	00	10
7EC0	42	42	2C	10	2C	62	62	00	B0
7EC8	42	42	2C	10	18	18	18	00	08
7ED0	7E	42	02	7E	60	62	7E	00	80
7ED8	3C	30	30	30	30	30	3C	00	68
7EE0	00	60	30	18	0C	06	00	00	BA
7EE8	3C	0C	0C	0C	0C	0C	3C	00	B4
7EF0	00	32	4C	00	00	00	00	00	7E
7EF8	00	00	00	00	00	00	7E	7E	FC
7F00	08	18	30	00	00	00	00	00	50
7F08	00	00	7A	46	62	62	7E	00	02
7F10	40	40	7E	42	62	62	7E	00	82
7F18	00	00	7E	40	60	60	7E	00	FC
7F20	02	02	7E	42	62	62	7E	00	06
7F28	00	00	7E	42	7E	60	7E	00	1C
7F30	0C	10	38	10	18	18	18	00	AC
7F38	00	00	7A	46	62	7E	02	7E	20
7F40	40	40	7C	42	62	62	62	00	64
7F48	10	00	10	10	18	18	18	00	78
7F50	08	00	08	08	18	18	18	70	D0
7F58	40	40	44	48	70	68	64	00	48
7F60	10	10	10	10	18	18	18	00	88
7F68	00	00	7C	4A	6A	6A	6A	00	04
7F70	00	00	7C	42	62	62	62	00	E4
7F78	00	00	7E	42	62	62	7E	00	02
7F80	00	00	7E	42	62	7E	40	40	20
7F88	00	00	7E	42	62	7E	02	02	A4
7F90	00	00	7C	42	60	60	60	00	DE
7F98	00	00	7E	40	7E	06	7E	00	C0
7FA0	10	10	38	10	18	18	18	0E	BE
7FA8	00	00	42	42	62	62	7E	00	C6
7FB0	00	00	42	42	64	68	30	00	80
7FB8	00	00	42	42	6A	6A	34	00	8C
7FC0	00	00	42	4C	30	6C	62	00	8C
7FC8	00	00	42	42	62	7E	02	7E	E4
7FD0	00	00	7E	02	7E	60	7E	00	DC
7FD8	0C	18	18	30	18	18	0C	00	A8
7FE0	18	18	18	00	18	18	18	00	90
7FE8	30	18	18	0C	18	18	30	00	CC
7FF0	00	18	3C	66	00	00	00	00	BA
7FF8	FF	FF	FF	FF	FF	FF	FF	FF	F8

```
7CC0   F5        u     PUSH  AF
7CC1   79        y     LD    A,C
7CC2   FE20      β     CP    20H
7CC4   3807      8     JR    C,$+09H
7CC6   FE80      β     CP    80H
7CC8   3003      0     JR    NC,$+05H
7CCA   C680      F     ADD   A,80H
7CCC   4F        O     LD    C,A
7CCD   F1        q     POP   AF
7CCE   C3E430    Cd0   JP    30E4H
7CD1   00              NOP
7CD2   21C07C    !§ö   LD    HL,7CC0H
7CD5   11A043      C   LD    DE,43A0H
7CD8   011100          LD    BC,0011H
7CDB   EDB0      m0    LDIR
7CDD   21A043    ! C   LD    HL,43A0H
7CE0   221E40    " §   LD    (401EH),HL
7CE3   21007D    ! ü   LD    HL,7D00H
7CE6   1100F5      u   LD    DE,0F500H
7CE9   010003          LD    BC,0300H
7CEC   EDB0      m0    LDIR
7CEE   AF        /     XOR   A
7CEF   CDB03F    M0?   CALL  3FB0H
7CF2   C36600    Cf    JP    0066H
```

Ein Screen-Printer

Sicher haben Sie sich gefragt, wie die vielen Bildschirmausdrucke, die in diesem Buch zu finden sind, erzeugt wurden.

Das folgende Maschinenspracheprogramm erzeugt einen solchen Bildschirmausdruck auf dem STAR DP 510/515. Dabei sind sowohl Ausdrucke von Texten, als auch von FGR-Grafiken möglich. Außerdem erkennt das Programm, welcher Zeichensatz für die Zeichen von 128 bis 255 angewählt wurde.
Nach dem Einschalten Ihres Rechners geben Sie als MEM SIZE 46330 ein.* Dann laden Sie den "Basic-Monitor" (oder jeden anderen Monitor, sofern dieser nicht die Adressen über B500H belegt) und geben das Programm mit dem E-Befehl ein. Wenn Sie sicher sind, daß Sie alles fehlerfrei eingegeben haben, schreiben Sie den Screen-Printer mit dem Befehl WB500 BFFF BE40 auf Kassette. Zum Schluß geben Sie ein:
GBE40.

Der Screen-Printer meldet sich mit:
<SCREEN PRINTER>
(c) 1983 von TCS

COLOUR BASIC
READY
>

Nun können Sie jederzeit durch Druck auf die Tasten <CTRL> und <P> einen Ausdruck des Bildschirms machen. Da dieses Programm den Tastatur-DCB anzapft, funktioniert das Programm immer, solange es nicht überschrieben wird oder die Tastatur-Treiber-Adresse verändert wird.

Hier nun ein Hex-Dump (mit Prüfsummen) des Screen-Printer:

* (Für dieses Programm muß Ihr Colour-Genie auf 32 K RAM erweitert sein.)

```
B500 30 FF FF FF FF FF FF FF   29
B508 FF FF FF FF FF FF FF FF   F8
B510 FF FF FF FF FF FF FF FF   F8
B518 FF FF FF FF FF FF FF FF   F8
B520 FF FF FF FF FF FF FF FF   F8
B528 FF FF FF FF FF FF FF FF   F8
B530 FF FF FF FF FF FF FF FF   F8
B538 FF FF FF FF FF FF FF FF   F8
B540 FF FF FF FF FF FF FF FF   F8
B548 FF FF FF FF FF FF FF FF   F8
B550 FF FF FF FF FF FF FF FF   F8
B558 FF FF FF FF FF FF FF FF   F8
B560 FF FF FF FF FF FF FF FF   F8
B568 FF FF FF FF FF FF FF FF   F8
B570 FF FF FF FF FF FF FF FF   F8
B578 FF FF FF FF FF FF FF FF   F8
B580 FF FF FF FF FF FF FF FF   F8
B588 FF FF FF FF FF FF FF FF   F8
B590 FF FF FF FF FF FF FF FF   F8
B598 FF FF FF FF FF FF FF FF   F8
B5A0 FF FF FF FF FF FF FF FF   F8
B5A8 FF FF FF FF FF FF FF FF   F8
B5B0 FF FF FF FF FF FF FF FF   F8
B5B8 FF FF FF FF FF FF FF FF   F8
B5C0 FF FF FF FF FF FF FF FF   F8
B5C8 FF FF FF FF FF FF FF FF   F8
B5D0 FF FF FF FF FF FF FF FF   F8
B5D8 FF FF FF FF FF FF FF FF   F8
B5E0 FF FF FF FF FF FF FF FF   F8
B5E8 FF FF FF FF FF FF FF FF   F8
B5F0 FF FF FF FF FF FF FF FF   F8
B5F8 FF FF FF FF FF FF FF FF   F8
B600 00 00 00 00 00 00 00 00   00
B608 10 10 10 10 10 00 10 00   60
B610 28 28 28 00 00 00 00 00   78
B618 28 28 7C 28 7C 28 28 00   C0
B620 10 3C 50 38 14 78 10 00   70
B628 60 64 08 10 20 4C 0C 00   54
B630 20 50 50 20 54 48 34 00   B0
B638 10 10 10 00 00 00 00 00   30
B640 08 10 20 20 20 10 08 00   90
B648 20 10 08 08 08 10 20 00   78
B650 10 54 38 10 38 54 10 00   48
B658 00 10 10 7C 10 10 00 00   BC
B660 00 00 00 00 10 10 20 00   40
B668 00 00 00 7C 00 00 00 00   7C
B670 00 00 00 00 00 00 10 00   10
B678 00 04 08 10 20 40 00 00   7C
B680 38 44 4C 54 64 44 38 00   FC
B688 10 30 10 10 10 10 10 00   90
B690 38 44 04 38 40 40 7C 00   B4
B698 7C 04 08 18 04 44 38 00   20
B6A0 18 28 48 7C 08 08 08 00   1C
B6A8 7C 40 78 04 04 44 38 00   B8
B6B0 1C 20 40 78 44 44 38 00   B4
B6B8 7C 04 04 08 10 20 40 00   FC
B6C0 38 44 44 38 44 44 38 00   B8
B6C8 38 44 44 3C 04 08 70 00   78
B6D0 00 00 10 00 10 00 00 00   20
B6D8 00 00 10 00 10 10 20 00   50
```

```
B6E0 08 10 20 40 20 10 08 00   B0
B6E8 00 00 7C 00 7C 00 00 00   F8
B6F0 20 10 08 04 08 10 20 00   74
B6F8 38 44 04 08 10 00 10 00   A8
B700 38 44 54 5C 58 40 3C 00   00
B708 10 28 44 44 7C 44 44 00   C4
B710 78 44 44 78 44 44 78 00   78
B718 38 44 40 40 40 44 38 00   B8
B720 78 44 44 44 44 44 78 00   44
B728 7C 40 40 70 40 40 7C 00   68
B730 7C 40 40 70 40 40 40 00   2C
B738 38 44 40 5C 44 44 38 00   D8
B740 44 44 44 7C 44 44 44 00   14
B748 38 10 10 10 10 10 38 00   C0
B750 7C 04 04 04 04 44 38 00   08
B758 44 48 50 60 50 48 44 00   18
B760 40 40 40 40 40 40 7C 00   FC
B768 44 6C 54 54 44 44 44 00   24
B770 44 64 54 4C 44 44 44 00   14
B778 38 44 44 44 44 44 38 00   C4
B780 78 44 44 78 40 40 40 00   38
B788 38 44 44 44 54 48 34 00   D4
B790 78 44 44 78 50 48 44 00   54
B798 38 44 40 38 04 44 38 00   74
B7A0 7C 54 10 10 10 10 10 00   20
B7A8 44 44 44 44 44 44 38 00   D0
B7B0 44 44 44 28 28 10 10 00   3C
B7B8 44 44 44 54 54 54 28 00   F0
B7C0 44 44 28 10 28 44 44 00   70
B7C8 44 44 28 10 10 10 10 00   F0
B7D0 7C 44 08 10 20 44 7C 00   B8
B7D8 78 60 60 60 60 60 78 00   D0
B7E0 00 40 20 10 08 04 00 00   7C
B7E8 3C 0C 0C 0C 0C 0C 3C 00   B4
B7F0 10 28 44 00 00 00 00 00   7C
B7F8 00 00 00 00 00 00 00 7C   7C
B800 20 10 08 00 00 00 00 00   38
B808 00 00 38 04 3C 44 3C 00   F8
B810 40 40 78 44 44 44 78 00   3C
B818 00 00 38 44 40 44 38 00   38
B820 04 04 3C 44 44 44 3C 00   4C
B828 00 00 38 44 7C 40 38 00   70
B830 18 20 70 20 20 20 20 20   48
B838 00 00 3C 44 44 3C 04 38   3C
B840 40 40 78 44 44 44 44 00   08
B848 10 00 30 10 10 10 38 00   A8
B850 08 00 08 08 08 08 48 30   A0
B858 40 40 48 50 60 50 48 00   10
B860 30 10 10 10 10 10 38 00   B8
B868 00 00 68 54 54 54 54 00   B8
B870 00 00 58 64 44 44 44 00   88
B878 00 00 38 44 44 44 38 00   3C
B880 00 00 78 44 44 78 40 40   F8
B888 00 00 3C 44 44 3C 04 04   08
B890 00 00 58 60 40 40 40 00   78
B898 00 00 3C 40 38 04 78 00   30
```

B8A0	10	10	38	10	10	10	10	08	A0
B8A8	00	00	44	44	44	4C	34	00	4C
B8B0	00	00	44	44	44	28	10	00	04
B8B8	00	00	44	44	54	54	28	00	58
B8C0	00	00	44	28	10	28	44	00	E8
B8C8	00	00	44	44	44	3C	04	38	44
B8D0	00	00	7C	08	10	20	7C	00	30
B8D8	18	20	20	40	20	20	18	00	F0
B8E0	10	10	10	00	10	10	10	00	60
B8E8	30	08	08	04	08	08	30	00	84
B8F0	00	00	32	4C	00	00	00	00	7E
B8F8	FF	FF	FF	FF	FF	FF	FF	FF	F8
B900	FF	FF	FF	07	07	07	07	07	20
B908	FF	FF	FF	E0	E0	E0	E0	E0	5D
B910	07	07	07	07	07	FF	FF	FF	20
B918	E0	E0	E0	E0	E0	FF	FF	FF	5D
B920	1C	1C	FC	FC	FC	FC	1C	1C	60
B928	38	38	3F	3F	3F	3F	38	38	DC
B930	3C	3C	3C	FF	FF	FF	00	00	B1
B938	00	00	FF	FF	FF	3C	3C	3C	B1
B940	00	00	FF	FF	FF	00	00	00	FD
B948	80	80	80	80	80	80	80	80	00
B950	01	01	01	01	01	01	01	01	08
B958	FF	00	00	00	00	00	00	00	FF
B960	00	00	00	00	00	00	00	FF	FF
B968	FF	81	81	81	81	81	81	FF	04
B970	38	38	38	38	38	38	38	38	C0
B978	00	00	00	18	18	00	00	00	30
B980	3C	3C	3C	3C	3C	3C	3C	3C	E0
B988	00	00	FF	FF	FF	FF	00	00	FC
B990	FC	FC	C0	C0	C0	C0	C0	C0	78
B998	3F	3F	3F	3F	03	03	03	03	08
B9A0	C0	C0	30	3C	3C	30	C0	C0	D8
B9A8	03	03	0C	3C	3C	0C	03	03	9C
B9B0	C0	C0	F0	F0	F0	F0	C0	C0	C0
B9B8	03	03	0F	0F	0F	0F	03	03	48
B9C0	00	00	3C	3C	3C	3C	00	00	F0
B9C8	0C	0C	3C	3C	3C	3C	30	30	68
B9D0	00	00	FC	FC	3F	3F	00	00	76
B9D8	C0	C0	C0	C0	C0	FC	FC	FC	B4
B9E0	03	03	03	03	3F	3F	3F	3F	08
B9E8	00	00	18	18	3C	3C	C3	C3	2E
B9F0	C3	C3	3C	3C	18	18	00	00	2E
B9F8	00	00	00	00	3C	3C	FF	FF	76
BA00	FF	FF	3C	3C	00	00	00	00	76
BA08	13	7E	7E	7E	7E	7E	7E	00	07
BA10	FF	FF	C0	C0	C0	C0	FF	FF	FC
BA18	3C	3C	3C	3C	3C	3C	3C	3C	E0
BA20	FF	FF	C3	C3	C3	C3	C3	C3	90
BA28	C3	C3	C3	C3	C3	C3	FF	FF	90
BA30	3C	3C	FF	FF	3C	3C	3C	3C	66
BA38	24	24	24	E7	E7	24	24	24	A6
BA40	C3	C3	C3	C3	C3	C3	C3	C3	18
BA48	FF	FF	FF	FF	FF	FF	18	18	2A
BA50	18	18	FF	FF	FF	FF	FF	FF	2A
BA58	C0	C0	C0	C0	C0	C0	C0	C0	00

BA60	FF	FF	03	03	03	03	FF	FF	08
BA68	FF	FF	F0	F0	F0	F0	0F	0F	DC
BA70	0F	0F	F0	F0	F0	F0	F0	FF	CD
BA78	FC	FC	FC	FC	C3	C3	C3	C3	FC
BA80	C3	C3	C3	C3	FC	FC	FC	FC	FC
BA88	FC	FC	FC	FC	FC	FC	FC	FC	E0
BA90	C3	C3	C3	C3	3F	3F	3F	3F	08
BA98	1F	1F	1F	1F	63	63	63	63	08
BAA0	00	00	00	00	00	00	FF	FF	FE
BAA8	FF	FF	00	00	00	00	00	00	FE
BAB0	03	03	03	03	03	03	03	03	18
BAB8	FF	81	BD	A5	A5	BD	81	FF	C4
BAC0	E7	66	66	66	FF	FF	FF	FF	15
BAC8	C3	C3	3C	3C	3C	C3	C3	00	C0
BAD0	18	18	18	18	18	18	18	18	C0
BAD8	00	00	00	00	00	00	F8	F8	F0
BAE0	00	00	00	00	00	00	1F	1F	3E
BAE8	0F	0F	0F	0F	00	00	00	00	3C
BAF0	00	00	00	00	F0	F0	F0	F0	C0
BAF8	F0	F0	00	00	00	00	00	00	E0
BB00	18	66	FF	18	24	42	81	81	FD
BB08	00	FF	00	FF	00	FF	00	FF	FC
BB10	18	18	18	FF	FF	18	18	18	8E
BB18	CC	CC	33	33	CC	CC	33	33	FC
BB20	AA	55	AA	55	AA	55	AA	55	FC
BB28	C3	66	3C	18	3C	66	C3	81	63
BB30	36	7F	7F	7F	3E	3E	1C	08	53
BB38	81	C3	DB	E7	DB	C3	81	00	25
BB40	08	1C	3E	7F	7F	36	08	3E	DC
BB48	08	1C	1C	2A	7F	2A	08	3E	59
BB50	FF	FF	FF	FF	FF	FF	FF	FF	F8
BB58	33	33	33	33	33	33	33	33	98
BB60	49	49	49	49	49	49	49	49	48
BB68	01	03	07	0F	1F	3F	7F	FF	F6
BB70	FF	FF	00	00	FF	FF	00	00	FC
BB78	C0	C0	C0	C0	C0	C0	C0	C0	00
BB80	00	00	FF	FF	00	00	FF	FF	FC
BB88	FF	FF	FF	FF	00	00	00	00	FC
BB90	00	00	00	00	FF	FF	FF	FF	FC
BB98	FF	FF	00	00	00	00	00	00	FE
BBA0	0F	0F	0F	0F	0F	0F	0F	0F	78
BBA8	F0	F0	F0	F0	F0	F0	F0	F0	80
BBB0	F0	F0	F0	F0	0F	0F	0F	0F	FC
BBB8	FF	00	FF	00	FF	00	FF	00	FC
BBC0	92	92	92	92	92	92	92	92	90
BBC8	00	00	00	FF	FF	00	00	00	FE
BBD0	00	00	00	00	00	00	FF	FF	FE
BBD8	03	03	03	03	03	03	03	03	18
BBE0	04	08	11	22	44	88	10	20	3B
BBE8	CC	CC	CC	CC	CC	CC	CC	CC	60
BBF0	20	10	88	44	22	11	08	04	3B
BBF8	01	03	06	0C	18	30	60	C0	7E
BC00	80	C0	E0	F0	F8	FC	FE	FF	01
BC08	18	18	18	18	18	18	18	18	C0
BC10	FF	FE	FC	F8	F0	E0	C0	80	01
BC18	C0	60	30	18	0C	06	03	01	7E

```
BC20 FF 7F 3F 1F 0F 07 03 01   F6
BC28 FF FF 7E 3C 00 00 00 00   B8
BC30 3C 7E FF FF FF FF 7E 3C   70
BC38 99 99 99 99 99 5A 3C 18   AB
BC40 18 3C 5A 99 18 18 18 18   A7
BC48 FF 81 81 99 99 81 81 FF   34
BC50 C0 C0 00 18 18 00 03 03   B6
BC58 3C 7E C3 C3 C3 C3 7E 3C   80
BC60 00 00 00 00 0F 08 08 08   27
BC68 00 00 00 00 F0 10 10 10   20
BC70 07 0F 1F 1F 1F 1F 0F 07   A8
BC78 08 08 08 0F 00 00 00 00   27
BC80 10 10 10 F0 00 00 00 00   20
BC88 E7 24 24 3C 66 C3 81 81   96
BC90 08 1C 3E 7F 7F 3E 1C 08   C2
BC98 24 42 A5 18 18 A5 42 24   46
BCA0 00 00 00 3C 7E FF FF FF   B7
BCA8 E0 F0 F8 F8 F8 F8 F0 E0   80
BCB0 00 00 00 18 18 00 00 00   30
BCB8 00 66 66 00 00 66 66 00   98
BCC0 FF C3 81 81 81 81 C3 FF   88
BCC8 C3 C3 00 18 18 00 C3 C3   3C
BCD0 66 FF FF 66 66 FF FF 66   94
BCD8 18 18 18 18 99 5A 3C 18   A7
BCE0 00 00 18 18 00 18 18 00   60
BCE8 10 20 40 FF FF 40 20 10   DE
BCF0 66 66 00 66 66 00 66 66   64
BCF8 08 04 02 FF FF 02 04 08   1A
BD00 DD 21 00 B5 21 1C 43 FE   31
BD08 80 38 16 FE C0 38 0A CB   99
BD10 5E 20 04 DD 21 00 F0 18   88
BD18 08 CB 66 20 04 DD 21 00   5B
BD20 F0 6F 26 00 29 29 29 EB   EB
BD28 DD 19 21 00 42 11 01 42   AD
BD30 01 08 00 36 00 ED B0 06   E2
BD38 08 16 80 C5 21 07 42 06   D3
BD40 08 1E 01 DD 7E 00 A3 28   4D
BD48 03 7A B6 77 CB 03 2B 10   B3
BD50 F2 C1 CB 0A DD 23 10 E3   7B
BD58 C9 00 00 00 00 00 00 00   C9
BD60 21 00 44 3A F9 42 47 C5   E6
BD68 3E 1B CD C0 BD 3E 4B CD   F9
BD70 C0 BD 3E 50 CD C0 BD 3E   93
BD78 00 CD C0 BD 06 50 CD C0   2D
BD80 BD 10 FB 06 28 7E C5 D5   0E
BD88 E5 CD 00 BD 3E 1B CD C0   55
BD90 BD 3E 4B CD C0 BD 3E 08   D6
BD98 CD C0 BD 3E 00 CD C0 BD   D2
BDA0 06 08 21 00 42 7E CD C0   7C
BDA8 BD 23 10 F9 E1 D1 C1 23   7F
BDB0 10 D3 3E 0D CD C0 BD C1   39
BDB8 05 C2 67 BD C9 00 00 00   B4
BDC0 F5 3E 07 D3 F8 3E 7F D3   95
BDC8 F9 3E 0F D3 F8 DB F9 E6   CB
BDD0 EF FE 2F 20 EC 3E 07 D3   40
BDD8 F8 3E 7F D3 F9 3E 0E D3   A0
BDE0 F8 F1 D3 F9 F5 3E 07 D3   C2
```

```
BDE8 F8 3E FF D3 F9 3E 0F D3   21
BDF0 F8 3E 00 D3 F9 3E 0F D3   22
BDF8 F8 3E 01 D3 F9 F1 C9 00   BD
BE00 3E 1B CD C0 BD 3E 41 CD   EF
BE08 C0 BD 3E 08 CD C0 BD C3   D0
BE10 60 BD 3A 80 F8 FE 10 C0   9D
BE18 3A 04 F8 FE 01 C0 C5 D5   8F
BE20 E5 DD E5 21 32 BE E5 3A   D7
BE28 1C 43 CB 6F CA 00 BE C3   E4
BE30 00 BF DD E1 E1 D1 C1 C9   B9
BE38 E5 21 E3 03 E3 C3 12 BE   62
BE40 21 38 BE 22 16 40 21 20   D0
BE48 BF CD A7 28 21 00 B5 C3   F4
BE50 02 01 00 00 00 00 00 00   03
BE58 00 00 00 00 00 00 00 00   00
BE60 DD 21 00 48 3A 12 43 47   1C
BE68 C5 06 28 21 00 42 DD 7E   B1
BE70 00 E6 C0 0F 0F 0F 0F 0F   F1
BE78 0F CD CC BE DD 7E 00 E6   A7
BE80 30 0F 0F 0F 0F CD CC BE   C3
BE88 DD 7E 00 E6 0C 0F 0F CD   38
BE90 CC BE DD 7E 00 E6 03 CD   9B
BE98 CC BE 3E 1B CD C0 BD 3E   6B
BEA0 4B CD C0 BD 3E 0C CD C0   6C
BEA8 BD 3E 00 CD C0 BD C5 06   10
BEB0 0C 21 00 42 7E CD C0 BD   37
BEB8 23 10 F9 C1 DD 23 10 AB   A8
BEC0 C1 3E 0D CD C0 BD 10 A0   06
BEC8 C9 00 00 00 B7 20 0A 36   E0
BED0 00 23 36 00 23 36 00 23   D5
BED8 C9 FE 01 20 0A 36 A0 23   EB
BEE0 36 A0 23 36 A0 23 C9 FE   B9
BEE8 02 20 0A 36 50 23 36 A0   AB
BEF0 23 36 50 23 C9 36 F0 23   DE
BEF8 36 F0 23 36 F0 23 C9 00   5B
BF00 3E 1B CD C0 BD 3E 41 CD   EF
BF08 C0 BD 3E 04 CD C0 BD C3   CC
BF10 60 BE 00 00 00 00 00 00   1E
BF18 00 00 00 00 00 00 00 00   00
BF20 3C 53 43 52 45 45 4E 20   1C
BF28 50 52 49 4E 54 45 52 3E   62
BF30 0D 28 63 29 20 31 39 38   83
BF38 33 20 76 6F 6E 20 54 43   5D
BF40 53 0D 0D 00 00 00 00 00   6D
BF48 00 00 00 00 00 00 00 00   00
BF50 00 00 00 00 00 00 00 00   00
BF58 00 00 00 00 00 00 00 00   00
BF60 00 00 00 00 00 00 00 00   00
BF68 00 00 00 00 00 00 00 00   00
BF70 00 00 00 00 00 00 00 00   00
BF78 00 00 00 00 00 00 00 00   00
BF80 00 00 00 00 00 00 00 00   00
BF88 00 00 00 00 00 00 00 00   00
BF90 00 00 00 00 00 00 00 00   00
BF98 00 00 00 00 00 00 00 00   00
```

```
BFA0 00 00 00 00 00 00 00 00    00
BFA8 00 00 00 00 00 00 00 00    00
BFB0 00 00 00 00 00 00 00 00    00
BFB8 00 00 00 00 00 00 00 00    00
BFC0 00 00 00 00 00 00 00 00    00
BFC8 00 00 00 00 00 00 00 00    00
BFD0 00 00 00 00 00 00 00 00    00
BFD8 00 00 00 00 00 00 00 00    00
BFE0 00 00 00 00 00 00 00 00    00
BFE8 00 00 00 00 00 00 00 00    00
BFF0 00 00 00 00 00 00 00 00    00
BFF8 00 00 00 00 00 00 00 00    00
```

Anhang A: Basic Einfachtokens

128	END	129	FOR
130	RESET	131	SET
132	CLS	133	CMD
134	RANDOM	135	NEXT
136	DATA	137	INPUT
138	DIM	139	READ
140	LET	141	GOTO
142	RUN	143	IF
144	RESTORE	145	GOSUB
146	RETURN	147	REM
148	STOP	149	ELSE
150	TRON	151	TROFF
152	DEFSTR	153	DEFINT
154	DEFSNG	155	DEFDBL
156	LINE	157	EDIT
158	ERROR	159	RESUME
160	OUT	161	ON
162	OPEN	163	FIELD
164	GET	165	PUT
166	CLOSE	167	LOAD
168	MERGE	169	NAME
170	KILL	171	LSET
172	RSET	173	SAVE
174	SYSTEM	175	LPRINT
176	DEF	177	POKE
178	PRINT	179	CONT
180	LIST	181	LLIST
182	DELETE	183	AUTO
184	CLEAR	185	CLOAD
186	CSAVE	187	NEW
188	TAB(	189	TO
190	FN	191	USING
192	VARPTR	193	USR
194	ERL	195	ERR
196	STRING$	197	INSTR
198	CHECK	199	TIME$
200	MEM	201	INKEY$
202	THEN	203	NOT
204	STEP	205	+
206	-	207	*
208	/	209	[
210	AND	211	OR
212	>	213	=
214	<	215	SGN
216	INT	217	ABS
218	FRE	219	INP
220	POS	221	SQR
222	RND	223	LOG
224	EXP	225	COS
226	SIN	227	TAN
228	ATN	229	PEEK
230	CVI	231	CVS
232	CVD	233	EOF
234	LOC	235	LOF
236	MKI$	237	MKS$
238	MKD$	239	CINT
240	CSNG	241	CDBL
242	FIX	243	LEN
244	STR$	245	VAL
246	ASC	247	CHR$
248	LEFT$	249	RIGHT$
250	MID$	251	'

Colour-Basic Doppeltokens

255	128	COLOUR	255	129	FCOLOUP
255	130	KEYPAD	255	131	JOY
255	132	PLOT	255	133	FGR
255	134	LGP	255	135	FCLS
255	136	PLAY	255	137	CIPCLE
255	138	SCALE	255	139	SHAPE
255	140	NSHAPE	255	141	XSHAPE
255	142	PAINT	255	143	CPOINT
255	144	NPLOT	255	145	SOUND
255	146	CHAR	255	147	RENUM
255	148	SWAP	255	149	FKEY
255	150	CALL	255	151	VERIFY
255	152	BGRD	255	153	NBGRD

(Die Tokens sind die der neuen Basic-ROMs,
die seit April '83 eingebaut sind.)

Folgendes Programm erzeugt eine aequivalente Liste auf dem Bildschirm:

```
10 CLS:C=-1:A=127:B=3:CHAR4
20 PRINT"Basic Einfachtokens":X=&H1650
30 IFPEEK(X)>127THEN50
40 PRINTCHR$(PEEK(X)AND127);:X=X+1:GOTO30
50 B=B+1:C=C+1:IFC=124THEN90
60 IFB<48THEN80
70 GOSUB160:CLS:B=0
80 A=A+1:PRINT:COLOUR1:PRINT$(INT(B/2)*40)+(A-INT(A/2)*2)*20,A;" ";:CO
LOUR5:PRINTCHR$(A);"  ";:COLOUR2:GOTO40
90 GOSUB160:CLS:B=3:C=-1:A=127:X=&H3930
100 PRINT"Colour-Basic Doppeltokens"
110 IFPEEK(X)>127THEN130
120 PRINTCHR$(PEEK(X)AND127);:X=X+1:GOTO110
130 IFA=129THENPRINT"R";
140 B=B+1:C=C+1:IFC=26THENEND
150 A=A+1:PRINT:COLOUR1:PRINT$(INT(B/2)*40)+(A-INT(A/2)*2)*20,"255";A;
" ";:COLOUR5:PRINTCHR$(A);"  ";:COLOUR2:GOTO120
160 PRINT$960,"Bitte druecken Sie <RETURN>";:INPUTR$:RETURN
```

Beachten Sie, daß ein Teil der Befehle erst im Disk-Basic benutzt werden (z B TIME $)
Im normalen Basic haben diese Befehle keine Funktion und fuhren zu "?SN ERROR"

Anhang B:

Dezimal/Hexadezimal/ASCII-Tabelle

Dez	=	Hex	Z.	Dez	=	Hex	Z.	Dez	=	Hex	Z.
0	=	00H		1	=	01H		2	=	02H	
3	=	03H		4	=	04H		5	=	05H	
6	=	06H		7	=	07H		8	=	08H	
9	=	09H		10	=	0AH		11	=	0BH	
12	=	0CH		13	=	0DH		14	=	0EH	
15	=	0FH		16	=	10H		17	=	11H	
18	=	12H		19	=	13H		20	=	14H	
21	=	15H		22	=	16H		23	=	17H	
24	=	18H		25	=	19H		26	=	1AH	
27	=	1BH		28	=	1CH		29	=	1DH	
30	=	1EH		31	=	1FH		32	=	20H	
33	=	21H	!	34	=	22H	"	35	=	23H	#
36	=	24H	$	37	=	25H	%	38	=	26H	&
39	=	27H	'	40	=	28H	(	41	=	29H	)
42	=	2AH	*	43	=	2BH	+	44	=	2CH	,
45	=	2DH	-	46	=	2EH	.	47	=	2FH	/
48	=	30H	0	49	=	31H	1	50	=	32H	2
51	=	33H	3	52	=	34H	4	53	=	35H	5
54	=	36H	6	55	=	37H	7	56	=	38H	8
57	=	39H	9	58	=	3AH	:	59	=	3BH	;
60	=	3CH	<	61	=	3DH	=	62	=	3EH	>
63	=	3FH	?	64	=	40H	@	65	=	41H	A
66	=	42H	B	67	=	43H	C	68	=	44H	D
69	=	45H	E	70	=	46H	F	71	=	47H	G
72	=	48H	H	73	=	49H	I	74	=	4AH	J
75	=	4BH	K	76	=	4CH	L	77	=	4DH	M
78	=	4EH	N	79	=	4FH	O	80	=	50H	P
81	=	51H	Q	82	=	52H	R	83	=	53H	S
84	=	54H	T	85	=	55H	U	86	=	56H	V
87	=	57H	W	88	=	58H	X	89	=	59H	Y
90	=	5AH	Z	91	=	5BH	[	92	=	5CH	\
93	=	5DH	]	94	=	5EH	^	95	=	5FH	_
96	=	60H	`	97	=	61H	a	98	=	62H	b
99	=	63H	c	100	=	64H	d	101	=	65H	e
102	=	66H	f	103	=	67H	g	104	=	68H	h
105	=	69H	i	106	=	6AH	j	107	=	6BH	k
108	=	6CH	l	109	=	6DH	m	110	=	6EH	n
111	=	6FH	o	112	=	70H	p	113	=	71H	q
114	=	72H	r	115	=	73H	s	116	=	74H	t
117	=	75H	u	118	=	76H	v	119	=	77H	w
120	=	78H	x	121	=	79H	y	122	=	7AH	z
123	=	7BH	{	124	=	7CH	¦	125	=	7DH	}
126	=	7EH	~	127	=	7FH	█	128	=	80H	▜
129	=	81H	▛	130	=	82H	▟	131	=	83H	▙
132	=	84H	┫	133	=	85H	┣	134	=	86H	┻

Dez		Hex	Z.	Dez		Hex	Z.	Dez		Hex	Z.
135	=	87H		136	=	88H	▬	137	=	89H	|
138	=	8AH	|	139	=	8BH	‾	140	=	8CH	_
141	=	8DH	□	142	=	8EH	▮	143	=	8FH	·
144	=	90H	▮	145	=	91H	▬	146	=	92H	┏
147	=	93H		148	=	94H		149	=	95H	
150	=	96H		151	=	97H		152	=	98H	▪
153	=	99H		154	=	9AH		155	=	9BH	┗
156	=	9CH	┛	157	=	9DH		158	=	9EH	
159	=	9FH		160	=	A0H		161	=	A1H	
162	=	A2H		163	=	A3H	▮	164	=	A4H	
165	=	A5H		166	=	A6H		167	=	A7H	
168	=	A8H		169	=	A9H		170	=	AAH	
171	=	ABH	▮	172	=	ACH		173	=	ADH	
174	=	AEH		175	=	AFH		176	=	B0H	
177	=	B1H	■	178	=	B2H		179	=	B3H	
180	=	B4H	_	181	=	B5H	‾	182	=	B6H	▮
183	=	B7H		184	=	B8H		185	=	B9H	
186	=	BAH	▮	187	=	BBH	_	188	=	BCH	
189	=	BDH	▪	190	=	BEH	▪	191	=	BFH	
192	=	C0H		193	=	C1H	≡	194	=	C2H	✚
195	=	C3H		196	=	C4H	▒	197	=	C5H	X
198	=	C6H	♥	199	=	C7H		200	=	C8H	♠
201	=	C9H		202	=	CAH		203	=	CBH	
204	=	CCH		205	=	CDH		206	=	CEH	
207	=	CFH	▮	208	=	D0H	≡	209	=	D1H	▬
210	=	D2H	▬	211	=	D3H	‾	212	=	D4H	▮
213	=	D5H	▮	214	=	D6H		215	=	D7H	≡
216	=	D8H	|||	217	=	D9H	—	218	=	DAH	_
219	=	DBH	▮	220	=	DCH	//	221	=	DDH	||
222	=	DEH	\\\\	223	=	DFH	/	224	=	E0H	◣
225	=	E1H	▮	226	=	E2H	◤	227	=	E3H	\\
228	=	E4H	◥	229	=	E5H		230	=	E6H	●
231	=	E7H	Ψ	232	=	E8H	↑	233	=	E9H	⊡
234	=	EAH		235	=	EBH	◘	236	=	ECH	
237	=	EDH	┐	238	=	EEH	◖	239	=	EFH	└
240	=	F0H	┘	241	=	F1H		242	=	F2H	♦
243	=	F3H		244	=	F4H		245	=	F5H	◗
246	=	F6H	·	247	=	F7H	::	248	=	F8H	□
249	=	F9H		250	=	FAH	♣	251	=	FBH	↓
252	=	FCH	:	253	=	FDH	←	254	=	FEH	::
255	=	FFH	→								

Folgendes Programm erzeugt ein aequivalentes Listing auf dem Bildschirm:

```
10 CLS:B=3
15 PRINT"Dez   Hex Z. Dez   Hex Z. Dez   Hex Z."
20 FORA=0TO255
30 IFA<32THENA$=" "ELSEA$=CHR$(A)
40 A1$=STR$(A):A1$=RIGHT$(A1$,LEN(A1$)-1)
50 IFLEN(A1$)<3THENA1$=STRING$(3-LEN(A1$),32)+A1$
60 A1$=A1$+" = "
70 A1=INT(A/16):A2=AAND15
80 IFA1<10THENA1$=A1$+CHR$(48+A1)ELSEA1$=A1$+CHR$(55+A1)
90 IFA2<10THENA1$=A1$+CHR$(48+A2)ELSEA1$=A1$+CHR$(55+A2)
100 A1$=A1$+"H "+A$
110 PRINT$INT(B/3)*40+(B-INT(B/3)*3)*13,;
120 B=B+1
130 COLOUR3:PRINTMID$(A1$,1,3);:COLOUR1:PRINTMID$(A1$,4,3);:COLOUR6:PR
INTMID$(A1$,7,3);:COLOUR2:PRINTMID$(A1$,10,2);
140 IFB<69THENNEXTA
150 IFINKEY$=""THEN150ELSEB=0:CLS:GOTO140
```

Anhang C:

Colour-Genie Software



****

Verzeichnis der verfügbaren Colour-Genie-Software

****

1.) HILFSPROGRAMME

COLOUR-COMPILER: Dieses Programm ist extrem wertvoll. Es bietet die Möglichkeit, Basicprogramme in Maschinenspracheprogramme zu übersetzten.
Der wesentliche Vorteil: Ihr Programm läuft nach Compilierung c.a. 40 mal (!) schneller.
Dies geschieht interaktiv, d.h. Basicprogramm, Maschinenprogramm und Compiler stehen gleichzeitig im Speicher, was ein sehr schnelles Arbeiten ermöglicht. Dabei können c.a. 9.5 kByte Basicprogramm verarbeitet werden. Verzichtet man auf die hochauflösende Grafik, steht noch mehr Speicher zur Verfügung.
Voraussetzung: Ihr COLOUR-GENIE MUSS AUF 32K RAM ERWEITERT SEIN. Die meisten Colour-Basic-Befehle werden vom Colour-Compiler unterstutzt. Die wesentlichste Einschränkung ist, daß der Colour-Compiler nur Ganzzahlen (Integers) verarbeitet.
Preis: 69.- DM

COLOUR-MONITOR I: Ein Maschinensprachemonitor mit starkem Befehlssatz, z.B. Disassemlieren, Ascii/Hex-Dump, Bander laden/schreiben, Bytes suchen, Speicher editieren/verschieben/relozieren, Hex/Dezimal-Umwandlung u.a.m.
Preis: 39.- DM

COLOUR-ASSEMBLER: Mit diesem Programm konnen Maschinenspracheprogramme auf komfortable Weise entwickelt werden. Das Programmieren geschieht mit Z80-Mnemonics, Labels etc.
Der Assembler erzeugt dann das Maschinenspracheprogramm, das zur Probe auch direkt ausgefuhrt werden kann.
Eine weitere Besonderheit des Assemblers ist, daß der Assembler-Text im ganz normalen Basic-Modus erzeugt wird, so daß alle Basic-Befehle wie EDIT, LLIST, CSAVE vom Assembler ausgenutzt werden.
Preis: 69.- DM

SOUND-EDITOR (auch im Handbuch "Colour Basic leicht gelernt" aufgelistet) : Ein nützliches Hilfsprogramm zur Programmierung des PSG-ICs (Sound-Chip). Sie editieren die PSG-Register auf dem Bildschirm und der entsprechende Ton wird gleichzeitig ausgegeben. Abschließend gibt das Programm alle PSG-Registerinhalte in dezimaler Schreibweise aus.
Preis: 25.- DM

ZEICHEN-EDITOR (auch im Handbuch "Colour Basic leicht gelernt" aufgelistet) : Dieses Programm ermoglicht es, 64 der 128 frei definierbaren Zeichen auf dem Bildschirm ubersichtlich zu editieren. Es konnen auch Grafiken, wie z.B eine Schreibschrift, erstellt werden. Abschließend werden die definierten Zeichen in ein Basicprogramm geschrieben, was diese fur den späteren Gebrauch oder zur Entwicklung eigener Programme wieder definiert. So wird die umstandliche Handhabung der Programmierung der definierbaren Zeichen umgangen.
Preis: 25.- DM

GRAFIK-EDITOR: Wenn Sie Ihr Colour-Genie auf 32K RAM aufgerustet haben, konnen Sie dieses Programm einsetzten, denn der "Grafik-Editor" hat eine Länge von 26000 Bytes!
Er bietet die Moglichkeit, Grafiken im FGR-Modus mit einer Vielzahl von leistungsstarken Kommandos zu kreieren.
Abschließend wird ein Basic-Programm erzeugt werden, das, eingebunden in ein eigenes Programm, das Bild blitzschnell wieder auf den Bildschirm bringt. Sie konnen also auch komplexe Grafiken sehr schnell fertigstellen und sie hinterher beliebig verwenden.
Preis: 69.- DM

SHAPER: Die Handhabung des Basicbefehls "SHAPE" ist recht umständlich. Hier bringt der "Shaper" Abhilfe: Mit wenigen Tastendrücken erzeugen Sie eine Figur, die abschließend im Foramt der "Shape-Table" abgespeichert wird.
Preis: 25.- DM

COLROT: Ein sehr nutzliches Grafikprogramm, für alle die mit großen Texten Aufmerksamkeit erregen wollen (z.B. im Schaufenster). Colrot erzeugt Laufschriften mit bis zu 512 Zeichen Länge, Titel und Fußschrift, Inversdarstellung, Intermezzo u.v.a.m.
Ein komfortabler Editor ermoglicht einfaches Arbeiten.
Preis: 69.- DM

BASICODE: Dieses Programm macht ihr Colour-Genie Basicode-kompatibel ! Basicode ist das Standard-Basic, in dem z.B. das WDR-Fernsehen Programme austrahlt. Eine Hardwareanderung ist nicht notig.
Preis: 25.- DM

PLOTTER: Plotter ist ein anspruchsvolles Mathematikprogramm, das beliebige Funktionen zeichnet, Wertetabellen erstellt u.s.w.
Preis: 39.- DM

ZEICHENEDITOR+: Dieser Editor, in Maschinensprache geschrieben, ermoglicht es alle 128 Zeichen sehr schnell und komfortabel zu editieren. Dabei gibt es sogar Befehle um Kreise zu zeichnen, Zeichen zu kopieren/duplizieren/rotieren/invertieren u.v.a.m.
Preis: 39.- DM

BASIC+5: Dieses Programm erweitert das normale Basic um 5 Befehle: Im Grafikmodus konnen beliebige Texte dargestellt werden, Rechteckflachen konnen gemalt werden, mit "SAVE" konnen Machinenspracheprogramme gesichert und mit "LOAD" komfortabel geladen werden. Ferner steht ein Eingabebefehl zur Verfugung, der an beliebige Speicherzellen schreibt.
Preis: 39.- DM

ZEICHENSÄTZE: Dieses Programm läßt Sie den vorhandenen Zeichensatz für alle ASCII-Zeichen wahlweise gegen einen von acht neuen, interessanten Zeichensätzen ersetzen.
Für dieses Programm muß Ihr Colour-Genie mit 32K RAM ausgerüstet sein.
Preis: 25.- DM

SCREEN-PRINTER: (Auch in dem Buch "Das Colour-Genie-Buch 1" aufgelistet.) Auf dieses Programm haben die Besitzer des STAR-Druckers DP 510 bzw. DP 515 sicher gewartet !
Es ermöglicht es Ihnen, den Bildschirm jederzeit komplett auf Ihren Drucker auszugeben, egal ob Sie im FGR- oder im LGR-Modus sind. Definierte und feste Grafikzeichen werden ebenfalls mit ausgedruckt. Auch für dieses Programm brauchen Sie 32K RAM und natürlich einen STAR DP 510 / DP 515 Drucker.
Preis: 39.- DM

2.) SPIEL- u. GRAFIKPROGRAMME

INVASION AUS DEM WELTRAUM: Ein Maschinenspracheprogramm, das die Sound- u. Grafikmoglichkeiten des Colour-Genies voll ausnutzt. Ihre Aufgabe ist es, einen Pulk von Invasoren abzuwehren, der sich der Erde nahert.
Die Invasoren werden dabei immer schneller und gefahrlicher.
Preis: 39.- DM

PUNKTEJAGD: Bei diesem Spiel geht es darum, alle Punkte in einem Labyrint aufzusammeln, bevor Sie von einem Wachter eingeholt werden. Schnelle Grafik und Ton durch Maschinensprache.
Preis: 25.- DM

WURM: Unser derzeit schnellstes Action-Spiel. Ein Wurm kommt von oben den Bildschirm herab und versucht Sie zu vernichten. Dabei hat er die Spinne, die Fliege und die Ente als Helfer. Kein Spiel fur ruhige Stunden!
Preis: 39.- DM

BREAK OUT: Bei diesem Video-Spiel muß eine Mauer mit Ihrem Ball zerstort werden. Dabei konnen verschieden Schwierigkeitsgrade vorgewahlt werden. Hohe Geschwindigkeit durch Maschinenspracheprogrammierung.
Preis: 39.- DM

DEMOPROGRAMM: Hiermit konnen Sie besser als mit der mit Ihrem Colour-Genie z.Zt. mitgelieferten englischen Demokassette die Moglichkeiten Ihres Colour-Genies demonstrieren.
Preis: 25.- DM

ANDROMEDA: Ein erstklassiges dreidimensionales Weltraumspiel mit ansprechender Grafik. Feindliche Raumschiffe kommen auf Sie zu; vernichten Sie diese, bevor sie Ihnen wertvolle Energie abgesaugt haben. Ist die Zeit abgelaufen, so eilt Ihnen Ihre Mutterbasis zu Hilfe; aber das Auftanken will auch gelernt sein.
Preis: 39.- DM

MAU-MAU: Endlich haben Sie einen Spielpartner, der nicht wutend die Karten wegwirft, wenn er am verlieren ist. Dies durfte wohl aber auch selten der Fall sein, vielmehr wahrscheinlich ist es, daß Sie den Computer vor Wut ausschalten, da ihr Computer Dank einer hervorragenden Taktik auf Sieg programmiert ist.
(Er schummelt nicht !!) Gespielt wird nach den ublichen Regeln.
Preis: 25.- DM

HEKTIK: Stellen Sie sich vor, Sie waren in einem Neubau mit 6 Geschossen, bei dem die Ebenen durch Leitern verbunden sind. Jetzt kommen Ihnen von oben Verfolger entgegen. Sie mußen sich ihnen stellen, denn es gibt keinen Fluchtweg. Graben Sie an strategisch wichtigen Stellen Locher in den Boden, um so Ihre Verfolger auszuschalten.
Preis: 39.- DM

METEOR: Ein Super-Action-Spiel im Grafik-Modus Ihres Colour-Genies. Sie befinden sich in einem Meteoritenfeld und werden von Meteoriten, Sternen und schiessenden Ufos bedrängt.
Das Programm zeichnet sich durch sehr schnelle Grafik und gute Toneffekte aus.
Preis: 69.- DM

MOTTEN: Bei diesem Videospiel müßen Sie Kolonnen von Motten bekämpfen, die in gefährlichen Sturzflügen anfliegen und dabei auch noch schiessen.
Preis: 39.- DM

PANIK: Ein Programm für 2 Spieler. Man muß versuchen, den Gegner einzumauern. Wählbare Geschwindigkeit macht "Mauer" entweder zu einem Reaktions- oder zu einem Strategiespiel.
Das Programm kann wahlweise mit Joysticks oder mit der Tastatur bedient werden.
Preis: 25.- DM

TAUSENDFUß: Hier sollen Sie einen Wurm so steuern, daß er nur Futter aber kein Gift frißt. Dazu kommen noch Kraftfutter und Gegengift. Ein sehr unterhaltsames Geschicklichkeitsspiel für die ganze Familie. Bei Spielbeginn können viele verschiedene Geschwindigkeiten und Schwierigkeitsstufen vorgewählt werden.
Preis: 39.- DM

COLOUR-SCHACH: Jetzt können Sie gegen Ihr Colour-Genie auch Schach spielen. Colour-Schach bietet 7 verschiedene Spielstärken, Aufzeichnung eines laufenden Spieles auf Kassette. Andern von Stellungen, einen Demonstrationsmodus und anderes mehr. Die Figuren werden auf ansprechende Weise grafisch dargestellt.
Preis: 69.- DM

EXNIMROID: Ein Denkspiel, abgeleitet vom bekannten Nim-Spiel. Es geht darum, aus mehreren Haufchen den letzten Spielstein zu nehmen. Sie spielen gegen den Computer.
Preis: 25.- DM

KINGS: Eine Regierungssimulation, bei der Sie fur eine Dauer von 8 Jahren eine Insel regieren sollen. Dabei kann nur der kluge Einsatz aller Faktoren zu einem Erfolg fuhren. Haben Sie Ihre Amtszeit uberlebt, werden Ihre Taten ausgewerted, und Sie konnen dann auch weiter regieren.
Preis: 25.- DM

MAUSI: Ein lustiges Videospiel, bei dem Sie als Maus im stromendem Regen Kase holen sollen, ohne naß zu werden.
Preis: 25.- DM

MAMPFMAN: Ahnlich wie unser Spiel "Punktejagd", nur daß Sie hier von mehreren Gespenstern verfolgt werden, die Sie unter bestimmten Bedingungen jedoch auch fressen konnen.
Preis: 39.- DM

COLOUR-KONG: Das absolute Videospiel ! Retten Sie Ihre Freundin vor dem wilden Affen. Mehrere Ebenen, super Grafik und Musik. Auf dem Band finden Sie Versionen fur 32K und 16K RAM
Preis: 69.- DM

PANZERKAMPF: Ein Spiel fur zwei Personen (JOYSTICKS und 32K RAM erforderlich). Liefern Sie sich mit Ihrem Gegner spannende Panzerschlachten ! Das Programm verfugt uber 3 verschiedene Gelande mit Minen etc.
Preis: 69.- DM

EIS: Ein spannendes, zugleich jedoch auch recht schwieriges Actionspiel mit strategischen Elementen. Sie befinden sich in einem Kuhlraum und mußen Eisblocke so ans Rutschen bringen, daß die gefahrlichen Schneemonster vernichtet werden.
Preis: 39.- DM

EAGLE: Vernichten Sie verschiedene Schwadronen von Vogeln, bevor Sie versuchen das große Mutterschiff zu zerstoren.
Preis: 39.- DM

GAME of LIFE: Dieses bekannte Programm simuliert das Wachstum von Bakterienkulturen. Die Regeln werden im Programm erklart und einige interessante Figuren sind fest abgespeichert.
Preis: 25.- DM

SAUG: Ein lustiges Videospiel, bei dem Sie aus einem unterirdischen Labyrint Termiten ansaugen mußen, wobei Sie sich vor giftigen, roten Termiten huten mußen.
Das Spiel kann auch mit Joysticks gesteuert werden.
Preis: 39.- DM

EXREVERSIC: Spielen Sie Reversi gegen Ihr Colour-Genie !
Dabei kann zwischen mehreren Spielstufen gewahlt werden.
Das Spielbrett wird auf ansprechende Weise graphisch dargestellt.
Preis: 39.- DM

BANG-BANG: Zwei Cowboys begegnen sich in der Prarie und liefern sich ein erbittertes Duell. Fur dieses Spiel sind Joysticks erforderlich.
Preis: 39.- DM

NETZO: Versuchen Sie mit Ihrem Pinsel alle Flachen auszumalen, ohne von Ihren Gegnern erwischt zu werden.
Preis: 39.- DM

DEATH-TRAP, ein dreidimensionales Grafik-Abenteuerspiel (Adventure):
Ein Programm zum wahnsinnig werden! Sie befinden sich in einem Labyrinth mit über 1100 Räumen, in dem es von gefährlichen Einwohnern und Gegenständen wimmelt. Finden Sie die wichtigen Gegenstände und wenden Sie sie richtig an, um aus dem Labyrinth zu entkommen. Zwei Vorraussetzungen müssen allerdings erfüllt sein: Ihr Colour-Genie muß 32K RAM haben, und Sie müssen elementare Englischkenntnisse besitzen, da die Sie Kommandos in Form von englischen Sätzen eingeben.
Preis: 69.- DM

TCS-CHOPPER: Die Geschmäcker sind verschieden - aber dies ist wohl das beste Videospiel, das es z.Zt. für das Colour-Genie gibt! Dreizehn Ihrer Kameraden sind in einem von vier riesigen Labyrinthen ausgesetzt worden. Versuchen Sie nun, diese mit Ihrem Hubschrauber zu retten! Aber Vorsicht, Ihre Freunde werden scharf bewacht!
32K RAM sind auch hier notwendig.
Preis: 69.- DM

DOPPEL-WURM: Vielleicht kennen Sie unser beliebtes Spiel "Tausendfuß". "Doppel-Wurm" ist sehr ähnlich, allerdings spielen hier zwei Spieler gegeneinander, was die Spannung natürlich erhöht. Achtung: Nur für Joystick-Besitzer !
Preis: 39.- DM

MADTREE: Ein schwieriges Videospiel: Sie sind eine fleißige Biene und müssen eine Reihe von Blumen ständig bestäuben, damit diese nicht eingehen. Doch diese Blumen sind undankbar und gefährden Sie.
Preis: 39.- DM

DIG-BOY: In Ihrem unterirdischen Reich kämpfen Sie gegen Drachen und andere Untiere. Locken Sie diese unter einen der wackligen Felsen, um sie zu zerschmettern.
Dieses Spiel ist sehr empfehlenswert, da es eine Menge Abwechslung bietet.
Preis: 39.- DM

EMPIRE: Für alle, die nicht nur reine Aktion wollen, ist "Empire" das ideale Spiel. Dieses Spiel kann mit bis zu sechs Spielern gleichzeitig gespielt werden, der Computer spielt ebenfalls mit. Jeder Spieler hat ein kleines Reich, das nun durch Handel, Politik, Krieg u.s.w. zu Wohlstand kommen soll. Dabei stehen Sie mit den anderen Nationen ständig in Beziehung, sei es durch Handel oder durch Krieg.
Ansprechende Grafik macht das Programm noch interessanter.
Ihr Colour-Genie muß für "Empire" auf 32K RAM aufgerüstet sein.
Preis: 69.- DM

3.) MUSIKPROGRAMME

MUSIK: Dieses Programm spielt festprogrammierte Musikstücke mit verblüffender Qualität.
Preis: 25.- DM

ORGEL: Spielen Sie Orgel auf Ihrem Colour-Genie !
Ihnen stehen zwei Manuale, einstellbares Delay und sogar eine Schlagzeugbegleitung zur Seite.
Preis: 25.- DM

COLOUR-SYNTHESIZER: Ein Programm, das man gesehen haben muß, um es überhaupt für möglich zu halten. Der Colour-Synthesizer macht aus Ihrem Colour-Genie einen vollwertigen 3-Kanal Synthesizer mit VCO, VCA, Hüllkurve, Schlagzeug...
Acht Einstellungen aller Regler können vorprogrammiert, auf Tastendruck abgerufen und auf Band gesichert werden.
Verblüffend ist auch die graphische Darstellung.
Preis: 69.- DM

COMPOSER: Mit diesem Programm können Sie sehr einfach eigene Musikstücke komponieren. Noten, Pausenzeichen u.s.w. werden grafisch dargestellt, können editiert und auf Band gespeichert werden.
Preis: 39.- DM

4.) LERNPROGRAMME (z.Zt. nur eines, demnächst mehr):

MATHEMATIK-LERNPROGRAMM: Interessant für Kinder bis in's Grundschulalter. Wahlweise können die vier Grundrechenarten geübt werden, wobei man zwischen verschiedenen Schwierigkeitsgraden wählen kann. Lustige grafische Gestaltung sorgt dafür, daß das Üben Spass macht.
Preis: 25.- DM

*** STAND DER TCS-SOFTWARELISTE.COLOUR-GENIE, VOM 12.8.1983 ***

In Vorbereitung: Colour-Text, Lernprogramme für Kinder, verbesserter Maschinensprache-Monitor und viele interessante Spiele !

BESTELLFORMULAR

Zur Bestellung von Colour-Genie-Programmen können Sie dieses Formular benutzten. Schicken Sie dieses an Ihren GENIE-Handler oder direkt an

TCS Computer GmbH
Kölnstraße 4
5205 St. Augustin 2
Tel.: 02241 / 28071

Bankverbindung:
Raiffeisenbank Menden BLZ 37069507 Konto-Nr. 1525

Der Versand erfolgt per Nachnahme oder Vorauskasse auf obiges Konto (dies natürlich nicht, wenn Sie bei Ihrem Händler bestellen).

Ich bitte um Zusendung folgender TCS Colour-Genie-Programme:

Anzahl	Name des Programms	Preis/DM	Gesamt

Gesamt: DM

Meine Adresse ist:

Name: ....

Straße: ....

Ort: ....

Datum: .... Unterschrift: ....

Kalle Braun · Jürgen Buchmüller · Frank Seger

Das Colour-Genie Buch 2

TCS
COMPUTER GMBH

Das Colour-Genie Buch 2

ISBN 3 88965 004 X

Vorwort

Lieber Colour-Genie Freund !

Nachdem unser 'Colour-Genie-Buch I' ein echter 'Renner' geworden ist, baten uns viele Colour-Genie-Benutzer, ein zweites Buch speziell für ihren Computer herauszubringen.

Gesagt - Getan:
Vor Ihnen liegt das 'Colour- Genie-Buch II'!

Auch dieses Mal finden Sie auf über 100 Seiten interessante Programme und Tricks; schauen Sie in das Inhaltsverzeichnis auf der nächsten Seite, um sich davon zu überzeugen.

Neben mehreren interessanten Spielen, darunter ein komplett erklärtes Maschinensprachespiel (!), Graphik- und Musikprogrammen, werden vor allen Dingen die Disk-Benutzer unter Ihnen wissenswertes in diesem Buch finden, z.B. eine schnelle Routine, mit der Disketten formatiert werden können u.v.m.

Wir wünschen Ihnen viel Spaß beim Hacken und sei der Tippfehlerteufel nicht allzusehr gegen Sie!

Bonn, im Dezember 1983

Kalle Braun
Jürgen Buchmüller
Frank Seger

Inhaltsverzeichnis

Einige allgemeine Hinweise

Bei allen Programmlistings in diesem Buch sind folgende Zeichenumwandlungen zu beachten:

'§' entspricht dem '@'
'A' entspricht dem '['

Die meisten Basic-Programme in diesem Buch sind 'entpackt', d.h. nach jedem Befehl folgt ein Leerzeichen, in jeder Zeile steht nur ein Befehl usw. Es ist natürlich nicht notig, alle diese Leerzeichen mit einzugeben, es erleichtert jedoch das Programmverstandnis und die Fehlersuche.

Es empfiehlt sich, zunachst das 'Colour-Genie Buch I' zu lesen, da dort wesentlich einfachere Dingen erklart und beschrieben werden. In diesem Buch werden sehr viele Probleme in Maschinensprache gelost, dies erfordert naturlich gewisse Kenntnisse des Rechners, die Sie sich zunachst aneignen sollten.

Um sich in die Maschinensprache des Colour-Genie einzuarbeiten, empfehlen wir Ihnen:

- 'Programmierung des Z80' von Rodnay Zaks, erschienen im Sybex-Verlag
- 'Das Colour-Genie ROM-Listing' von TCS

Ein Musikprogramm in Basic: Der 'BLUESMAKER'

Das folgende Basicprogramm erzeugt Melodien, die einem Blues nachempfunden sind.
Es sind dabei drei Grundakkorde vorgegeben, die in den DATA Zeilen 190 bis 210 stehen. Die erste Zahl entspricht der Oktave und die zweite der Tonhoehe, wie sie im PLAY Befehl benutzt werden. In Zeile 220 stehen die drei Töne für den Baß. Zeile 230 enthält codiert die Folge der Akkorde In Zeile 240 bis 260 stehen die Töne, die bei der Melodie zum jeweiligen Akkord gespielt werden können.
Die Melodie wird während des Programmlaufs zufällig aus den zu dem Akkord passenden Noten ausgewählt. Mit der Eingabe des Tempos bestimmen Sie die Geschwindigkeit der Baßbegleitung.

Programmlisting:

```
100 DEFINT A-Z
110 CLS:COLOUR5
120 A$="COLOUR GENIE Bluesmaker"
130 FORL=1TO13
140 COLOUR(LAND15)+1
150 PRINTTAB(L)A$
160 NEXTL
170 PRINT:PRINT
180 INPUT"Tempo (1...6) ";TE
190 DATA 5,2,6,2,12,2,6,2
200 DATA 1,3,2,3,9,3,2,3
210 DATA 2,3,3,3,4,3,3,3
220 DATA 1,2,4,2,5,2
230 DATA1,1,1,1,2,2,2,2,1,1,1,1,3,2,1,1
240 DATA 1,4,9,4,3,4,10,4,5,4,12,4,1,5
250 DATA 1,4,9,4,4,4,10,4,5,4,12,4,1,5
260 DATA 5,3,12,3,7,3,8,4,2,4,4,4,5,4
270 DIM BT(3,4),BO(3,4)
280 DIM LT(3),LO(3)
290 DIM L(16)
300 DIM JT(3,7),JO(3,7)
310 FOR X=1TO3:FOR Y=1TO4
320 READ BT(X,Y),BO(X,Y)
330 NEXT Y,X
340 FOR X=1TO3
350 READ LT(X),LO(X)
360 NEXT X
370 FOR X=1TO16
380 READ L(X)
390 NEXT X
400 FOR X=1TO3:FOR Y=1TO7
410 READ JT(X,Y),JO(X,Y)
420 NEXT Y,X
```

```
430 C=1:T=1:B=0
440 GOSUB 470
450 GOSUB 630
460 GOTO 440
470 IF B<>1 THEN 580
480 PLAY (1,1,0,0)
490 PLAY (2,1,0,0)
500 D=D+1:IF D<>1 THEN RETURN
510 D=0:B=0
520 T=T+1
530 IF T<>5 THEN RETURN
540 T=1
550 P=P+1:IF P=17 THEN P=1
560 C=L(P)
570 RETURN
580 D=D+1
590 IF D=TEMPO THEN B=1:D=0
600 PLAY (1,LO(C),LT(C),8)
610 PLAY (2,BO(C,T),BT(C,T),8)
620 RETURN
630 NT=NT+RND(3)-2
640 IF NT<1 THEN NT=1
650 IF NT>7 THEN NT=7
660 PLAY (3,JO(C,NT),JT(C,NT),9)
670 RETURN
```

Enterprise - ein strategisches Weltraumspiel

(Fur dieses Spiel muß Ihr Colour-Genie mit 32K RAM ausgerüstet sein. Das Programmlisting ist kommentiert, so daß wir Ihnen nur noch moglichst wenige Fehler beim Abtippen wunschen !)

SPIELERKLARUNG

1) Einleitung:

Enterprise ist ein Strategiespiel, bei dem es auf eine kluge Taktik ankommt. Man bewegt sich durch die Milchstraße auf der Suche nach Klingonen, die die friedlichen Planeten der Föderation mit Tod und Vernichtung uberziehen. Sie sind nun der Kommandant des Sternschiffs ENTERPRISE, dem einzig verfugbaren Kampfraumer in dem Teil der Galaxis, wo die Klingonen zuletzt geortet worden sind.
Unter Lebensgefahr mussen Sie die Klingonen bekampfen und vernichten, um die Planeten vor dem endgultig Untergang zu bewahren. Zu Ihrer Unterstützung sind in der Galaxis einige Raumbasen verteilt, an die Sie bei Bedarf andocken konnen. Die Enterprise wird dann aufgetankt und evtl. repariert. Wenn Sie andocken wollen, mussen Sie nur in denselben Raumsektor steuern, in der die Basis steht.

2) Kommandos:

Ein Kommando wird vom Computer dann erwartet, wenn auf dem Bildschirm 'BEFEHL ?' erscheint. Wenn Sie einen falschen Befehl eingeben, wird eine Zusammenfassung der gultigen Befehle ausgegeben.

NAV -> Steuern

Mit diesem Kommando konnen Sie die Enterprise durch die Galaxis bewegen. Die Galaxis ist in Quadraten und Sektoren unterteilt. Wollen Sie Klingonen bekampfen, so mussen sich diese in Ihrem Quadranten aufhalten. Um einen solchen Quadranten zu erreichen, geben Sie den Befehl NAV ein. Sie werden dann nach Richtung und Lange Ihres Kurses gefragt. Der Wert fur die Richtung liegt zwischen 1 und 8 und bedeutet folgendes:

```
      3
   4     2
 5    *    1
   6     8
      7
```

Die Eingabe von Richtungswerten erlaubt auch Kommazahlen damit eine genaue Steuerung moglich ist.
Die Kurslange wird durch den Warpfaktor festgelegt.
Die Zahl vor dem Komma gibt dabei die Anzahl der zu durchfliegenden Quadranten und die Nachkommazahl die der Sektoren an. Aber Vorsicht ist geboten ! Sollte ein Stern auf Ihrer Flugbahn liegen, so werden die Triebwerke automatisch abgeschaltet und die Energie fur den Flug geht verloren.

NAT -> Nahabtaster

Der Nahabtaster zeigt Ihnen ein Bild des Quadranten, in dem sich die ENTERPRISE aufhält, sowie einen verkürzten Statusreport. Ein Quadrant ist in 8 mal 8 Sektoren unterteilt.

Die Symbole bedeuten:

Enterprise	E
Klingone	K
Raumbasis	B
Stern	*

FAT -> Fernabtaster

Der Fernabtaster zeigt den Inhalt ddes eigenen und der umliegenden acht Quadranten an. Der dreistelligen Zahl entnehmen Sie:

1. Ziffer - Anzahl der Klingonen
2. Ziffer - Anzahl der Raumstationen
3. Ziffer - Anzahl der Sterne

PHA -> Phaser

Mit dem Phaser kann amn alle Klingonen im Quadranten mit der Energie beschießen, die man angibt. Ist die abgefeuerte Energiemenge zu knapp gewesen, so greifen die Klingonen ihrerseits die ENTERPRISE und können erheblichen Schaden anrichten (Status-Report beachten!).

TOR -> Torpedo

Sie feuern ein Torpedo ab, indem Sie wie beim Navigieren den Kurs angeben. Sollten nach dem Abschuß des Torpedos noch Klingonen vorhanden sein, so schießen diese ebenfalls auf die ENTERPRISE. Der Torpedokurs kann über COM, Option2 vom Bordcomputer errechnet werden.

SCH -> Schutzschirm

Dieser Befehl verändert die Energie der Schutzschirme. Wird die ENTERPRISE angegriffen, so sind diese Schirme das einzige Mittel einen Kampf überstehen. Ohne Energie auf den Schutzschirmen wäre Ihr Schiff beim geringsten Treffer vernichtet.

COM -> Computerabfrage

Nachdem Sie dieses Kommando eingegeben haben, wir die Frage nach einer Option gestellt. Diese Option muß eine Zahl zwischen 0 und 4 sein, die folgende Bedeutungen haben:

0: Zeigt alle bisher erfolgten Fernabtastungen. (Code siehe oben)

1: Gibt einen Statusreport

2: Torpedokursberechnung (Richtung/Länge)

3: Rechnet Kurs und Geschwindigkeit zur Sternbasis in Ihrem Quadranten uas. Der Warpfaktor beträgt nur 1/10 des Geschwindigkeitswertes.

4: Berechnet den Kurs zwischen zwei Punkten.

Lassen Sie sich durch die Vielfalt der Kommandos nicht verwirren. Nach einiger Übung wird Ihnen jeder Befehl in Fleisch und Blut übergehen. Etwaige Unklarheiten werden sicherlich im Spiel geklärt werden.

Programmlisting:

```
10 '    SUPER STAR TREK
20 '
30 RANDOM
40 CLEAR 600
50 COLOUR 3
60 CLS
70 PRINT"Der Weltraum.... unendliche Weiten....    "
80 COLOUR 5
90 PRINT"Wir schreiben das Jahr 2163 alter
100 PRINT"Zeitrechnung. Die ENTERPRISE ist mit
110 PRINT"ihrer ueber 400 Mann starken Besatzung
120 PRINT"Unterwegs um neue Welten zu erforschen.
130 PRINT"neues Leben und neue Zivilisation.
140 PRINT
150 COLOUR 11
160 PRINT"Sie dringt dabei in Galaxien vor, die
```

```
170 PRINT"nie ein Mensch zuvor gesehen hat !"
180 PRINT"Ihre Aufgabe als Captain der Enterprise
190 PRINT"ist es, Ihr Schiff und die Besatzung
200 PRINT"vor den Angriffen der feindlichen
210 PRINT"Klingonen zu schuetzen
220 PRINT
230 COLOUR 16
240 PRINT"Die Foederation wuenscht Ihnen
250 PRINT"                    viel Erfolg.
260 Z$="                              "
270 DIM G(8, 8), C(9, 2), K(3, 3), N(3), Z(8, 8), D(8)
280 T=( RND(20)+20)*100
290 T0=T
300 T9=RND(10)+25
310 D0=0
320 E=3000
330 E0=E
340 P=10
350 P0=P
360 S9=200
370 S=0
380 E9=0
390 K9=0
400 X$="basis"
410 X0$=" ist"
420 '  INIT ENTERPRISE POSITION
430 Q1=RND(8)
440 Q2=RND(8)
450 S1=RND(8)
460 S2=RND(8)
470 FOR I=1 TO 9
480 C(I, 1)=0
490 C(I, 2)=0
500 NEXT
510 C(3, 1)=-1
520 C(2, 1)=-1
530 C(4, 1)=-1
540 C(4, 2)=-1
550 C(5, 2)=-1
560 C(6, 2)=-1
570 C(1, 2)=1
580 C(2, 2)=1
590 C(6, 1)=1
600 C(7, 1)=1
610 C(8, 1)=1
620 C(9, 2)=1
630 C(8, 2)=1
640 FOR I=1 TO 8
```

```
650 D(I)=0
660 NEXT
670 A1$="NAVNATFATPHATORSCHZERCOM"
680 '  GALAXIS AUFBAUEN
690 '  K3=Klingonen  B3=Basis  S3=Stern
700 FOR I=1 TO 8
710 FOR J=1 TO 8
720 K3=0
730 Z(I, J)=0
740 R1=RND(0)
750 IF R1>.98 THEN K3=3: K9=K9+3: GOTO 780
760 IF R1>.95 THEN K3=2: K9=K9+2: GOTO 780
770 IF R1>.80 THEN K3=1: K9=K9+1
780 B3=0
790 IF RND(0)>.96 THEN B3=1: B9=B9+1
800 G(I, J)=K3*100+B3*10+RND(8)
810 NEXT
820 NEXT
830 IF K9>T9 THEN T9=K9+1
840 IF B9=<>0 THEN 900
850 IF G(Q1, Q2)<200 THEN G(Q1, Q2)=G(Q1, Q2)+100: K9=K9+1
860 B9=1
870 G(Q1, Q2)=G(Q1, Q2)+10
880 Q1=RND(8)
890 Q2=RND(8)
900 K7=K9
910 IF B9<>1 THEN X$="basen": X0$=" sind"
920 COLOUR RND(16)
930 PRINT # 930,"Druecke <RETURN>"
940 IF(PEEK(&HF840)AND 1)=0 THEN 920
950 CLS
960 COLOUR 16
970 PRINT"Das Oberkommando gibt Dir den Auftrag:"
980 PRINT
990 PRINT"Zerstoere";K9;"Klingonische Raumschiffe,"
1000 PRINT"die in Foederationsgebiet eingefallen"
1010 PRINT"sind.bevor sie das Hauptquartier"
1020 PRINT"angreifen koennen."
1030 COLOUR 7
1040 PRINT
1050 PRINT"Du hast bis STARDATE";T0+T9;"Zeit.also";T9
1060 PRINT"Einheiten"
1070 COLOUR 6
1080 PRINT
1090 PRINT"Es";X0$; B9;"Sternen";X$;" im eventuellen" '
1100 PRINT"Kampfgebiet.um Dich zu versorgen"
1110 '  QUADRANT BERECHNEN
```

```
1120 Z4=Q1
1130 Z5=Q2
1140 K3=0
1150 B3=0
1160 S3=0
1170 G5=0
1180 D4=RND(.5)
1190 Z(Q1, Q2)=G(Q1, Q2)
1200 IF Q1<1 OR Q2<1 OR Q1>8 OR Q2>8 THEN 1470
1210 GOSUB 4940
1220 PRINT
1230 IF T0<>T THEN 1320
1240 COLOUR 16
1250 PRINT
1260 PRINT"Deine Mission beginnt im galaktischen   Quadranten '";
1270 COLOUR 5
1280 PRINT G2$;
1290 COLOUR 16
1300 PRINT"'."
1310 GOTO 1330
1320 PRINT"Du kommst nach '";G2$;"'."
1330 PRINT
1340 K3=INT(G(Q1, Q2)*.01)
1350 B3=INT(G(Q1, Q2)*.1)-10*K3
1360 S3=G(Q1, Q2)-100*K3-10*B3
1370 IF K3=0 THEN 1430
1380 COLOUR 3
1390 PRINT"KAMPFGEBIET            CONDITION ROT"
1400 IF S>200 THEN 1430
1420 PRINT"SCHUTZSCHIRME GEFAEHRLICH SCHWACH"
1430 FOR I=1 TO 3
1440 K(I, 1)=0
1450 K(I, 2)=0
1460 NEXT
1465 COLOUR 16
1470 FOR I=1 TO 3
1480 K(I, 3)=0
1490 NEXT
1500 Q$=Z$+Z$+Z$+Z$+Z$+Z$+Z$+LEFT$(Z$, 17)
1510 '  POSITIONEN IN QUADRANTEN FESTLEGEN
1520 A$="<E>"
1530 Z1=S1
1540 Z2=S2
1550 GOSUB 4870
1560 IF K3<1 THEN 1670
1570 FOR I=1 TO K3
```

```
1580 GOSUB 4780
1590 A$="+K+"
1600 Z1=R1
1610 Z2=R2
1620 GOSUB 4870
1630 K(I, 2)=R2
1640 K(I, 1)=R1
1650 K(I, 3)=S9*(.5+RND(0))
1660 NEXT
1670 IF B3<1 THEN 1750
1680 GOSUB 4780
1690 A$=">B<"
1700 Z1=R1
1710 B4=R1
1720 Z2=R2
1730 B5=R2
1740 GOSUB 4870
1750 FOR I=1 TO S3
1760 GOSUB 4780
1770 A$=" * "
1780 Z1=R1
1790 Z2=R2
1800 GOSUB 4870
1810 NEXT
1820 GOSUB 5460
1830 IF S+E>10 THEN IF E>10 OR D(7)=0 THEN 1870
1840 PRINT
1850 PRINT"      SCHWERER  FEHLER "
1860 GOTO 4640
1870 INPUT"Befehl ";A$
1880 FOR I=1 TO 9
1890 IF LEFT$(A$, 3)<>MID$(A1$, 3*I-2, 3)THEN 1910
1900 ON I GOTO 2030, 1820, 3190, 3410, 3790, 4480, 6530, 7280
1910 NEXT
1920 PRINT
1930 PRINT"FOLGENDE EINGABEN SIND MOEGLICH"
1940 PRINT"  NAV =>  BEWEGEN"
1950 PRINT"  NAT =>  NAHABTASTER"
1960 PRINT"  FAT =>  FERNABTASTER"
1970 PRINT"  PHA =>  PHASER"
1980 PRINT"  TOR =>  TORPEDO"
1990 PRINT"  SCH =>  SCHUTZSCHIRME"
2000 PRINT"  ZER =>  SCHADENSMELDUNG"
2010 PRINT"  COM =>  COMPUTERANFRAGE"
2020 GOTO 1830
2030 '  FLUGKONTROLLE
2040 INPUT"Richtung ";C1
2050 IF C1=9 THEN C1=1
```

```
2060 IF C1<1 OR C1>9 THEN PRINT: PRINT" Ltn. Sulu:Falscher Kurs.Sir !"
: GOTO 1830
2070 X$="8"
2080 IF D(1)<0 THEN X$="0.2"
2090 PRINT"Warp Faktor (0-";X$;")";
2100 INPUT W1
2110 IF D(1)<0 AND W1>.2 THEN 2160
2120 IF W1>0 AND W1<=8 THEN 2190
2130 IF W1=0 THEN 1830
2140 PRINT
2150 PRINT"Cheftechniker Scott: Warp";W1;" ? Das     schaffen unsere M
aschinen nicht !"
2160 PRINT"Warptriebwerke beschaedigt !"
2170 PRINT"        Maximale Beschleunigung:0.2 Warp"
2180 GOTO 1830
2190 N=INT(W1*8+.5)
2200 IF E-N>=0 THEN 2280
2210 PRINT
2220 PRINT"Maschinenraum:Fuer Warp";W1;" reicht die   Energie nicht au
s !"
2230 IF S<N-E OR D(7)<0 THEN 1830
2240 PRINT
2250 PRINT"Schirmkontrolle:Schilde sind mit";S
2260 PRINT"Energieeinheiten geladen !"
2270 GOTO 1830
2280 '  BEWEGEN & FEUERN DER KLINGONEN
2290 FOR I=1 TO K3
2300 IF K(I, 3)=0 THEN 2400
2310 A$="   "
2320 Z1=K(I, 1)
2330 Z2=K(I, 1)
2340 GOSUB 4870
2350 GOSUB 4780
2360 K(I, 1)=Z1
2370 K(I, 2)=Z2
2380 A$="+K+"
2390 GOSUB 4870
2400 NEXT
2410 GOSUB 6330
2420 D1=0
2430 D6=W1
2440 IF W1>=1 THEN D6=1
2450 FOR I=1 TO 8
2460 IF D(I)>=0 THEN 2510
2470 D(I)=D(I)+D6
2480 IF D(I)>-.1 AND D(I)<0 THEN D(I)=-.1: GOTO 2510
2490 IF D(I)<0 THEN 2510
2500 IF D(I)<>1 THEN D1=1: R1=I: GOSUB 6060: PRINT"Schadensmeldung:";G
```

```
2$;" repariert"
2510 NEXT
2520 IF RND(0)>.2 THEN 2640
2530 R1=RND(8)
2540 IF RND(0)>=.6 THEN 2600
2550 D(R1)=D(R1)-( RND(5))
2560 PRINT"Schadensmeldung:";
2570 GOSUB 6060
2580 PRINT G2$;" beschaedigt"
2590 GOTO 2690
2600 D(R1)=D(R1)+RND(3)
2610 PRINT"Schadensmeldung:";
2620 GOSUB 6060
2630 PRINT" reparieren schreitet voran"
2640 '  ENTERPRISE WIRD BEWEGT
2650 A$="   "
2660 Z1=INT(S1)
2670 Z2=INT(S2)
2680 GOSUB 4870
2690 X1=C(C1, 1)+(C(C1+1, 1)-C(C1, 1))*(C1-INT(C1))
2700 X=S1
2710 Y=S2
2720 X2=C(C1, 2)+(C(C1+1, 2)-C(C1, 2))*(C1-INT(C1))
2730 Q4=Q1
2740 Q5=Q2
2750 FOR I=1 TO N
2760 S1=S1+X1
2770 S2=S2+X2
2780 IF S1<1 OR S1>=9 OR S2<1 OR S2>=9 THEN 3000
2790 S8=INT(S1)*24+INT(S2)*3-26
2800 IF MID$(Q$, S8, 2)="  " THEN 2850
2810 S1=INT(S1-X1)
2820 S2=INT(S2-X2)
2830 PRINT"Warptriebwerke schalteten sich wegen    schlechter Navigati
on in Sektor          ";S1;",";S2;" ab"
2840 GOTO 2880
2850 NEXT
2860 S1=INT(S1)
2870 S2=INT(S2)
2880 A$="<E>"
2890 Z1=INT(S1)
2900 Z2=INT(S2)
2910 GOSUB 4870
2920 GOSUB 6240
2930 T8=1
2940 IF W1<1 THEN T8=.1*INT(W1*10)
2950 T=T+T8
2960 IF T>T0+T9 THEN 4640
2970 '  IST ENTERPRISE ANGEDOCKT ?
2980 GOTO 1820
2990 '
```

```
3000 X=8*Q1+X+N*X1
3010 Y=8*Q2+Y+N*X2
3020 Q1=INT(X/8)
3030 Q2=INT(Y/8)
3040 S1=INT(X-Q1*8)
3050 S2=INT(Y-Q2*8)
3060 IF S1=0 THEN Q1=Q1-1: S1=8
3070 IF S2=0 THEN Q2=Q2-1: S2=8
3080 X5=0
3090 IF Q1<1 THEN X5=1: Q1=1: S1=1
3100 IF Q1>8 THEN X5=1: Q1=8: S1=8
3110 IF Q2<1 THEN X5=1: Q2=1: S2=1
3120 IF Q2>8 THEN X5=1: Q2=8: S2=8
3130 IF X5<>0 THEN PRINT: PRINT"Ltn. Uhura: Es ist uns verboten,die
  Galaxis zu verlassen.Die Triebwerke     sollen abgestellt werden !":
 PRINT"Cheftechniker Scott:Triebwerke gestoppt"
3135 PRINT" Cpt. Chekov: Wir sind im Sektor";S1;",";S2;"des Quadranten
";
3140 IF T>T0+T9 THEN 4640
3150 IF 8*Q1+Q2=8*Q4+Q5 THEN 2880
3160 T=T+1
3170 GOSUB 6240
3180 GOTO 1120
3190 '  FERNRADAR
3200 '
3210 IF D(3)<0 THEN PRINT"Fernradar faellt aus": GOTO 1830
3220 PRINT"Fernradar Auswertung fuer
Quadrant";Q1;",";Q2
3230 O1$="-------------------"
3240 PRINT O1$
3250 FOR I=Q1-1 TO Q1+1
3260 N(1)=-1
3270 N(2)=-2
3280 N(3)=-3
3290 FOR J=Q2-1 TO Q2+1
3300 IF I>0 AND I<9 AND J>0 AND J<9 THEN N(J-Q2+2)=G(I, J): Z(I, J)=G(
I, J)
3310 NEXT
3320 FOR L=1 TO 3
3330 PRINT": ";
3340 IF N(L)<0 THEN PRINTSTRING$(3,195);" ";: GOTO 3360
3350 PRINT RIGHT$(STR$(N(L)+1000), 3);" ";
3360 NEXT
3370 PRINT":"
3380 PRINT O1$
3390 NEXT
3400 GOTO 1830
3410 '  PHASER KONTROLLE
3420 '
3430 IF D(4)<0 THEN PRINT"Phaser sind inaktiv": GOTO 1830
```

```
3440 IF K3<0 THEN PRINT"IO Spock:Sensoren zeigen keine Klingonen in di
esem Quadranten": GOTO 1830
3450 IF D(8)<0 THEN PRINT"Fehler im Feuerleitcomputer"
3460 PRINT"Phaser im Ziel"
3470 PRINT"Verfuegbare Energie:";E;" Einheiten"
3480 INPUT"Schussenergie ";X
3490 IF X<=0 THEN 1830
3500 IF E-X<0 THEN 3470
3510 E=E-X
3520 IF D(7)<0 THEN X=X*RND(0)
3530 H1=INT(X/K3)
3540 FOR I=1 TO 3
3550 IF K(I, 3)<=0 THEN 3760
3560 H=INT((H1/SQR((K(I, 1)-S1)A2+(K(I, 2)-S2)A2))*( RND(0)+2))
3570 IF H>.15*K(I, 3)THEN 3600
3580 PRINT"Sensoren zeigen keinen Treffer"
3590 GOTO 3760
3600 K(I, 3)=K(I, 3)-H
3610 PRINT H;" Einheiten trafen Klingonen in":PRINT"Sektor";K(I, 1);",
";K(I, 2)
3620 IF K(I, 3)<=0 THEN PRINT"    *** KLINGONE ZERSTOERT ***": GOTO 36
60
3630 PRINT"   (Sensoren zeigen";K(I, 3)
3640 PRINT"    verbleibende Einheiten)"
3650 GOTO 3760
3660 K3=K3-1
3670 K9=K9-1
3680 Z1=K(I, 1)
3690 Z2=K(I, 2)
3700 A$="   "
3710 GOSUB 4870
3720 K(I, 3)=0
3730 G(Q1, Q2)=G(Q1, Q2)-100
3740 Z(Q1, Q2)=G(Q1, Q2)
3750 IF K9<=0 THEN 7830
3760 NEXT
3770 GOSUB 6330
3780 GOTO 1830
3790 '   PHOTONENTORPEDO KONTROLLE
3800 '
3810 IF P<=0 THEN PRINT"Keine Torpedos mehr verfuegbar": GOTO 1830
3820 IF D(5)<0 THEN PRINT"Agschusseinrichtung ausser Funktion": GOTO 1
830
3830 INPUT"Torpedokurs (1-9) ";C1
3840 IF C1=9 THEN C1=1
3850 IF C1<1 OR C1>9 THEN PRINT"Chekov:Falsche Kursdaten,Sir!": GOTO 1
830
3860 X1=C(C1, 1)+(C(C1+1, 1)-C(C1, 1))*(C1-INT(C1))
```

```
3870 E=E-2
3880 P=P-1
3890 X2=C(C1, 2)+(C(C1+1, 2)-C(C1, 2))*(C1-INT(C1))
3900 X=S1
3910 Y=S2
3920 PRINT"Torpedo fliegt"
3930 X=X+X1
3940 Y=Y+X2
3950 X3=INT(X+.5)
3960 Y3=INT(Y+.5)
3970 IF X3<1 OR X3>8 OR Y3<1 OR Y3>8 THEN 4450
3980 PRINT"                    ";X3;",";Y3
3990 A$="   "
4000 Z1=X
4010 Z2=Y
4020 GOSUB 5390
4030 IF Z3<>0 THEN 3930
4040 A$="+K+"
4050 Z1=X
4060 Z2=Y
4070 GOSUB 5390
4080 IF Z3=0 THEN 4190
4090 PRINT"*** Klingone zerstoert ***"
4100 K3=K3-1
4110 K9=K9-1
4120 IF K9<=0 THEN 7830
4130 FOR I=1 TO 3
4140 IF X3=K(I, 1)AND Y3=K(I, 2)THEN 4170
4150 NEXT
4160 I=3
4170 K(I, 3)=0
4180 GOTO 4370
4190 A$=" * "
4200 Z1=X
4210 Z2=Y
4220 GOSUB 5390
4230 IF Z3<>0 THEN PRINT"Stern bei";X3;",";Y3;" absorbierte Torpedo":
GOSUB 6330: GOTO 1830
4240 A$=">B<"
4250 Z1=X
4260 Z2=Y
4270 GOSUB 5390
4280 IF Z3=0 THEN 3820
4290 PRINT"*** Sternenbasis zerstoert ***"
4300 B3=B3-1
4310 B9=B9-1
4320 IF B9>0 OR K9>T-T0-T9 THEN 4350
4330 PRINT"Das wars,Kapitaen!!!"
4340 PRINT"Du bist hiermit Deines Komandos entzogenund zu 99 Sternenta
gen Strafarbeit auf  Cygnus 12 verdonnert !!!"
4350 PRINT"Das Flottenkommando sammelt Anklage-    punkte gegen Dich"
```

```
4360 D0=0
4370 Z1=X
4380 Z2=Y
4390 A$="   "
4400 GOSUB 4870
4410 G(Q1, Q2)=K3*100+B3*10+S3
4420 Z(Q1, Q2)=G(Q1, Q2)
4430 GOSUB 6330
4440 GOTO 1830
4450 PRINT"Torpedo ging am Ziel vorbei"
4460 GOSUB 6330
4470 GOTO 1830
4480 '   SCHIRMKONTROLLE
4490 IF D(7)<0 THEN PRINT"Schirmkontrolle ausgefallen": GOTO 1830
4500 PRINT"Verfuegbare Energie:";E+S
4510 INPUT"Wieviel Energie in die Schirme ";X
4520 IF X<0 OR S=X THEN PRINT"<Schirme unveraendert>": GOTO 1830
4530 IF X>E+S THEN PRINT"Schirmkontrolle:Wir sind nicht die      Schat
zkammer der Foederation": PRINT"    <Schirme unveraendert>": GOTO 1830
4540 E=E+S-X
4550 S=X
4560 PRINT"Deflektorraum:Schilde jetzt auf"
4570 PRINT INT(S);" Einheiten"
4580 GOTO 1830
4590 '   ENDE DES SPIELS
4600 CLS
4610 IF T>T0+T9 THEN PRINT"Deine Zeit ist ueberschritten,das Haupt-qua
rtier ist verloren und mit ihm die   Galaxis.Du Niete !!!!!": END
4620 PRINT"Heute ist der Sternentag";T
4630 GOTO 4680
4640 PRINT
4650 PRINT"       DIE ENTERPRISE IST ZERSTOERT"
4660 PRINT"   DIE FOEDERATION IST VERLOREN"
4670 GOTO 4620
4680 IF K9>1 THEN PRINT"Schade.es waren noch";K9;" Klingonische": PRIN
T"Schlachtschiffe uebrig" ELSE PRINT"So ein Mist,haettest Du das letzt
e": PRINT"Schiff doch noch erwischt.Pech!"
4690 PRINT"Das ist das Ende Deiner Mission !"
4700 PRINT
4710 PRINT
4720 IF B9=0 THEN END
4730 PRINT"Die Foederation braucht einen neuen     Kommandanten.also F
reiwillige vor !!!!!"
4740 PRINT
4750 INPUT"Tritt heran und tippe 'AYE' ";A$
4760 IF A$="AYE" THEN RUN
4770 END
4780 '   LEEREN PLATZ IM QUADRANTEN FINDEN
4790 R1=RND(8)
```

```
4800 A2=RND(8)
4810 A$="   "
4820 Z1=A1
4830 Z2=A2
4840 GOSUB 5390
4850 IF Z3=0 THEN 4790
4860 RETURN
4870 '
4880 S8=INT(Z2-.5)*3+INT(Z1-.5)*24+1
4890 IF LEN(A$)<>3 THEN PRINT"FEHLER": STOP
4900 IF S8=1 THEN Q$=A$+RIGHT$(Q$, 189): RETURN
4910 IF S8=190 THEN Q$=LEFT$(Q$, 189)+A$: RETURN
4920 Q$=LEFT$(Q$, S8-1)+A$+RIGHT$(Q$, 190-S8)
4930 RETURN
4940 '  NAME DES GEBIETS BESTIMMEN
4950 IF Z5<=4 THEN ON Z4 GOTO 4970, 4990, 5010, 5030, 5050, 5070, 5090
, 5110
4960 GOTO 5130
4970 G2$="Antares"
4980 GOTO 5120
4990 G2$="Rigel"
5000 GOTO 5120
5010 G2$="Procyon"
5020 GOTO 5120
5030 G2$="Wega"
5040 GOTO 5120
5050 G2$="Canopus"
5060 GOTO 5120
5070 G2$="Altair"
5080 GOTO 5120
5090 G2$="Sagittarius"
5100 GOTO 5120
5110 G2$="Pollux"
5120 GOTO 5290
5130 ON Z4 GOTO 5140, 5160, 5180, 5200, 5220, 5240, 5260, 5280
5140 G2$="Sirius"
5150 GOTO 5290
5160 G2$="Deneb"
5170 GOTO 5290
5180 G2$="Capella"
5190 GOTO 5290
5200 G2$="Beteigeuze"
5210 GOTO 5290
5220 G2$="Aldebaran"
5230 GOTO 5290
5240 G2$="Regulus"
5250 GOTO 5290
5260 G2$="Arcterus"
```

```
5270 GOTO 5290
5280 G2$="Spica"
5290 IF G5<>1 THEN ON Z5 GOTO 5310, 5330, 5350, 5370, 5310, 5330, 5350
, 5370
5300 RETURN
5310 G2$=G2$+" I"
5320 RETURN
5330 G2$=G2$+" II"
5340 RETURN
5350 G2$=G2$+" III"
5360 RETURN
5370 G2$=G2$+" IV"
5380 RETURN
5390 Z1=INT(Z1+.5)
5400 Z2=INT(Z2+.5)
5410 S8=(Z2-1)*3+(Z1-1)*24+1
5420 Z3=0
5430 IF MID$(Q$, S8, 3)<>A$THEN RETURN
5440 Z3=1
5450 RETURN
5460 '   ANFANGSBILD  UND NAHABTASTER
5470 '
5480 FOR I=S1-1 TO S1+1
5490 FOR J=S2-1 TO S2+1
5500 IF INT(I+.5)<1 OR INT(I+.5)>8 OR INT(J+.5)<1 OR INT(J+.5)>8 THEN
5560
5510 A$=">8<"
5520 Z1=I
5530 Z2=J
5540 GOSUB 5390
5550 IF Z3=1 THEN 5600
5560 NEXT
5570 NEXT
5580 DO=0
5590 GOTO 5670
5600 DO=1
5610 C$="DOCK"
5620 E=EO
5630 P=PO
5640 PRINT"Schirme werden fuers andocken abgebaut"
5650 S=0
5660 GOTO 5700
5670 IF K3>0 THEN C$="*ROT*": GOTO 5700
5680 C$="GRUEN"
5690 IF E<EO*.1 THEN C$="GELB "
5700 IF D(2))=0 THEN 5750
5710 PRINT
5720 PRINT"*** keine Nahabtastung moeglich ***"
```

```
5730 PRINT
5740 RETURN
5750 O1$=CHR$(223)+STRING$(16,211)+CHR$(227)
5760 O2$=CHR$(227)+STRING$(16,218)+CHR$(223)
5770 PRINT O1$
5780 FOR I=1 TO 8
5790 PRINTCHR$(207);
5800 FOR J=(I-1)*24+1 TO(I-1)*24+22 STEP 3
5810 PRINT" ";MID$(Q$, J+1, 1);
5820 NEXT
5830 ON I GOTO 5840, 5860, 5920, 5940, 5960, 5980, 6000, 6020
5840 PRINTCHR$(219);"   Sternentag";INT(T*10)*.1
5850 GOTO 6030
5860 PRINTCHR$(219);"   Condition  ";
5870 IF C$="*ROT*" THEN COLOUR 3:GOTO5890
5880 COLOUR 2
5890 PRINT C$
5900 COLOUR 16
5910 GOTO 6030
5920 PRINTCHR$(219);"   Quadrant   ";Q1;",";Q2
5930 GOTO 6030
5940 PRINTCHR$(219);"   Sektor     ";S1;",";S2
5950 GOTO 6030
5960 PRINTCHR$(219);"   Torpedos   ";INT(P)
5970 GOTO 6030
5980 PRINTCHR$(219);"   Energie    ";INT(E+S)
5990 GOTO 6030
6000 PRINTCHR$(219);"   Schirme    ";INT(S)
6010 GOTO 6030
6020 PRINTCHR$(219);"   Klingonen  ";INT(K9)
6030 NEXT
6040 PRINT O2$
6050 RETURN
6060 '  WAS WURDE ZERSTOERT
6070 ON R1 GOTO 6080, 6100, 6120, 6140, 6160, 6180, 6200, 6220
6080 G2$="Warp Triebwerk"
6090 RETURN
6100 G2$="Nahabtaster"
6110 RETURN
6120 G2$="Fernabtaster"
6130 RETURN
6140 G2$="Phaserkontrolle"
6150 RETURN
6160 G2$="Photonenroehren"
6170 RETURN
6180 G2$="Schadenskontrolle"
6190 RETURN
6200 G2$="Schirmkontrolle"
```

```
6210 RETURN
6220 G2$="Computer"
6230 RETURN
6240 '   ENERGIEVERBRAUCH
6250 '
6260 E=E-N-10
6270 IF E>=0 THEN RETURN
6280 PRINT"Zum beenden des Maneuvers muss Schirm-  energie abgezogen w
erden !"
6290 S=S+E
6300 E=0
6310 IF S<=0 THEN S=0
6320 RETURN
6330 '   KLINGONEN FEUERN
6340 IF K3<=0 THEN RETURN
6350 IF D0<>0 THEN PRINT"Schilde der Sternenbasis schuetzten     Enter
prise": RETURN
6360 FOR I=1 TO 3
6370 IF K(I, 3)<=0 THEN 6510
6380 H=INT((K(I, 3)/SQR((K(I, 1)-S1)Ä2+(K(I, 2)-S2)Ä2))*(2+RND(0)))
6390 S=S-H
6400 K(I, 3)=K(I, 3)/(3+RND(0))
6410 PRINT H;" Einheiten trafen Enterprise vom "
6420 PRINT"Sektor";K(I, 1);",";K(I, 2)
6430 IF S<=0 THEN 4590
6440 PRINT"<Schilde auf";S;" Einheiten gesunken>"
6450 IF H<20 THEN 6510
6460 IF RND(0)>.6 OR H/S<=.02 THEN 6510
6470 R1=RND(8)
6480 D(R1)=D(R1)-H/S-.5*RND(0)
6490 GOSUB 6060
6500 PRINT"Schadensmeldung: ";G2$;" zerstoert"
6510 NEXT
6520 RETURN
6530 '  SCHADENSMELDUNG
6540 IF D(6)>=0 THEN 6730
6550 PRINT"Schadensmeldung ausser Funktion"
6560 IF D0=0 THEN 1830
6570 D3=0
6580 FOR I=1 TO 8
6590 IF D(I)<0 THEN D3=D3+.1
6600 NEXT
6610 IF D3=0 THEN 1830
6620 PRINT
6630 D3=D3+D4
6640 IF D3>=1 THEN D3=.9
6650 PRINT"Techniker stehen bereit,um die Schaeden zu beheben"
6660 PRINT"Vorgesehene Zeit:";.01*INT(100*D3);" Sternentage"
```

```
6670 INPUT"Erlaubst Du die Reparatur (J/N) ";A$
6680 IF A$<>"J" THEN 1830
6690 FOR I=1 TO 8
6700 IF D(I)<0 THEN D(I)=0
6710 NEXT
6720 T=T+D3+.1
6730 PRINT
6740 PRINT
6750 PRINT"  Objekt                   Reparaturstadium"
6760 FOR R1=1 TO 8
6770 GOSUB 6060
6780 PRINT G2$; LEFT$(Z$, 30-LEN(G2$)); INT(D(R1)*100)*.01
6790 NEXT R1
6800 PRINT
6810 IF D0<>0 THEN 6570
6820 GOTO 1830
6830 '    PHOTONENTORPDATEN
6840 '
6850 IF K3<=0 THEN PRINT"Keine Klingonen in Reichweite !": GOTO 1830
6860 IF K3=1 THEN X$="zum" ELSE X$="zu den"
6870 PRINT"Von der Enterprise ";X$;" Klingonen"
6880 H8=0
6890 FOR I=1 TO 3
6900 IF K(I, 3)<=0 THEN 7220
6910 W1=K(I, 1)
6920 X=K(I, 2)
6930 C1=S1
6940 A=S2
6950 GOTO 7010
6960 PRINT"Entfernungsbestimmung:"
6970 PRINT"Du bist in Quadrant";Q1;",";Q2:PRINT"           Sektoren";S
1;",";S2:PRINT" Gebe ein
6980 '
6990 INPUT"Anfangskoordinate (X,Y) ";C1, A
7000 INPUT"    Endkoordinate (X,Y) ";W1, X
7010 X=X-A
7020 A=C1-W1
7030 IF X<0 THEN 7130
7040 IF A<0 THEN 7150
7050 IF X>0 THEN 7070
7060 IF A=0 THEN C1=5: GOTO 7080
7070 C1=1
7080 IF ABS(A)<=ABS(X)THEN 7110
7090 PRINT"Richtung:";C1+((( ABS(A)-ABS(X))+ABS(A))/ABS(A))
7100 GOTO 7200
7110 PRINT"Richtung:";C1+( ABS(A)/ABS(X))
7120 GOTO 7200
7130 IF A>0 THEN C1=3: GOTO 7160
7140 IF X<>0 THEN C1=5: GOTO 7080
```

```
7150 C1=7
7160 IF ABS(A)>=ABS(X)THEN 7190
7170 PRINT"Richtung:";C1+((( ABS(X)-ABS(A))+ABS(X))/ABS(X))
7180 GOTO 7200
7190 PRINT"Richtung:";C1+( ABS(X)/ABS(A))
7200 PRINT"Distanz :";SQR(XÄ2+AÄ2)
7210 IF H8=1 THEN 1830
7220 NEXT
7230 GOTO 1830
7240 '   KURS ZUR BASIS
7250 IF B3<>0 THEN PRINT"Von der Enterprise zur Basis:": W1=B4: X=B5:
GOTO 6930
7260 PRINT"Mr.Spock:Die Sensoren zeigen keine      Basis"
7270 GOTO 1830
7280 '  BUECHEREICOMPUTER
7290 IF D(8)<0 THEN PRINT"Computer inaktiv !": GOTO 1830
7300 INPUT"Computer ist bereit und wartet auf      Eingabe ";A
7310 IF A<0 THEN 1830
7320 PRINT
7330 H8=1
7340 ON A+1 GOTO 7450, 7730, 6830, 7240, 6960
7350 PRINT
7360 PRINT"      Abrufbare Funktionen :"
7370 PRINT"         0-Karte aller Abtastungen"
7380 PRINT"         1-Status Report"
7390 PRINT"         2-Photonentorpedodaten"
7400 PRINT"         3-Entfernung zur Basis"
7410 PRINT"         4-Entfernungsberechnung"
7430 PRINT
7440 GOTO 7300
7450 PRINT
7460 PRINT
7470 PRINT"Computeraufzeichnung der Galaxis ";Q1;",";Q2
7480 PRINT
7490 PRINT"      1   2   3   4   5   6   7   8"
7500 O1$="    "+STRING$(31,217)
7510 PRINT O1$
7520 FOR I=1 TO 8
7530 PRINT I;
7540 IF H8=0 THEN 7590
7550 FOR J=1 TO 8
7560 PRINT"F";
7570 IF Z(I, J)=0 THEN PRINTSTRING$(3,127);: GOTO 7590
7580 PRINT RIGHT$(STR$(Z(I, J)+1000), 3);
7590 NEXT
7600 PRINTCHR$(225):
7610 GOTO 7690
7620 Z4=I
7630 Z5=1
7640 J0=INT(15-.5*LEN(G2$))
7650 PRINT TAB(J0): G2$;
```

```
7660 Z5=5
7670 J0=INT(39-.5*LEN(G2$))
7680 PRINT TAB(J0); G2$;
7690 PRINT
7700 PRINT O1$
7710 NEXT
7720 GOTO 1830
7730 '   STATUS REPORT
7740 PRINT
7750 PRINT"  Status Report:"
7760 IF K9>1 THEN X$="en" ELSE X$=""
7770 PRINT"Noch";K9;" Klingon";X$;" uebrig"
7780 PRINT"Die Mission muss in";.1*INT((T9+T0-T)*10);" Sternentagen"
7790 PRINT"beendet werden"
7800 IF B9>1 OR B9=0 THEN X$="basen" ELSE X$="basis"
7810 PRINT"Es gibt";B9;" Sternen";X$
7820 GOTO 6530
7830 '   GEWONNEN
7840 PRINT
7850 PRINT
7860 PRINT
7870 PRINT"       DU HAST ALLE KLINGONEN VERNICHTET"
7880 PRINT"Dir gebuehrt der Dank der gesamten Milchstrasse!"
7890 PRINT"Die Foederation ernennt Dich zum"
7900 PRINT"                ADMIRAL"
7910 PRINT"und gibt Dir das Kommando ueber die";RND(10);"Raumflotte.da
mit Du auch weiterhin Klin-gonen jagst !"
7920 IF INKEY$ ="" THEN 7920
```

Run to Morlock's Tower

Auf den folgenden Seiten finden Sie das Listing eines Abenteuerspiels, bei dem Sie durch Verhandeln, Kämpfen, Forschen u.s.w. versuchen sollen, das Endziel (Eroberung des 'Morlock-Towers') zu erreichen.
Im Programmlisting finden Sie ab Programmzeile 7150 die Spielanleitung. Beachten Sie, daß das eigentliche Spiel in Englisch abläuft - die wichtigen Begriffe werden größtenteils in der Anleitung erklärt, im schlimmsten Fall müssen Sie halt mal ein Wörterbuch zur Hand nehmen.
Für dieses Programm muß Ihr Colour-Genie mit 32K RAM ausgerüstet sein.

Programmlisting:

```
10 COLOUR16
20 GOSUB 8160
30 CLS
40 PRINT @ 400,"BRAUCHEN SIE EINE ANLEITUNG (J/N)?";CHR$(14);
50 A$=INKEY$
60 IFA$="N" THEN 80
70 IFA$="J" THEN GOTO 7150ELSE GOTO 50
80 CLEAR 132:PRINTCHR$(15)
90 DIM D$(9), O$(17), D(32, 4), O(17, 6)
100 CLS
110 GOSUB 150
120 GOSUB 8800
130 GOSUB 7040
140 GOTO 220
150 DATA 7, 255 , 8, 16,  9, 16,  10, 16,  13, 16
160 RESTORE
170 FOR I=1 TO 5
180 READ XX, YY
190 SOUND XX,YY
200 NEXT
210 RETURN
220 IF INKEY$ ="" THEN GOTO 220
230 CLS
240 C=17408
250 O(17, 1)=RND(155)+100
260 SP=RND(6)
270 ST=RND(10)
280 CH=RND(6)
290 M=10
300 MC=6
310 PL=880
320 L=40
330 BQ=223
340 S1=223
350 D$="want to"
360 W$="which item?"
370 E$="don't have any!"
380 H$="how many?"
390 M$="don't have that many!"
400 Y$="you're wasting my time!"
```

```
410 R$="they attacked anyway!"
420 DATA67 CASTLE.82 RUIN.99 CAVE.84 TOWER.107 CITY.80 PILGRIMS.87 WIZ
ARDS,79 OGRES.83 SOLDIERS
430 DATASIEGE TOWERS,SMALL BOAT,GRAPPLING HOOKS,OIL LAMPS,AXES,SHOVELS
,ROPE,LEATHER JERKINS,CHAIN MAIL,PLATE ARMOR,SHIELDS,BROAD SWORDS,LONG
 BOWS,CROSSBOWS,RATIONS,LARGE KEY,GOLD COINS
440 DATA 1500, 2500,,150, 255,,15, 30,,5, 20,,10, 30,,10, 20,,15, 25,,
15, 40, 2, 25, 60, 4, 50, 100, 8, 3, 20, 2, 10, 50, 5, 11, 50, 5, 18,
40, 4, 4, 15,,1, 5,,,25, 1,
450 FOR I=1 TO 9
460 READ D$(I)
470 NEXT
480 FOR I=1 TO 17
490 READ O$(I)
500 NEXT
510 FOR I=1 TO 17
520 FOR J=2 TO 4
530 READ O(I, J)
540 NEXT J, I
550 DATA 67, 84, 99, 82, 67, 84, 99, 82, 67, 84, 99, 82, 67, 84, 99, 8
2, 67, 94, 99, 82, 67, 94, 99, 82, 80, 80, 87, 79, 83, 83
560 CLS
570 PRINT @ 216,"please wait"
580 COLOUR 6
590 PRINT @ 256,STRING$(11,211)
600 COLOUR 16
610 D(1, 1)=107
620 D(2, 1)=107
630 D(1, 2)=RND(6)
640 D(2, 2)=RND(6)+8
650 D(1, 3)=RND(30)+1
660 D(2, 3)=RND(30)+1
670 FOR I=3 TO 32
680 D(I, 4)=1
690 READ D(I, 1)
700 D(I, 2)=RND(14)
710 D(I, 3)=RND(30)+1
720 IF D(I, 2)=8 THEN 700
730 FOR J=1 TO I
740 IF D(I, 2)=D(J, 2)THEN IF D(I, 3)=D(J, 3)THEN IF J<>I THEN 700
750 NEXT J, I
760 FOR I=8 TO 14
770 IF RND(10)>5 THEN O(I, 1)=RND(8)
780 NEXT
790 O(15, 1)=RND(25)+15
800 A=RND(7)
810 B=RND(32)
820 SL=RND(384)+17500
830 GOSUB 6610
840 N=C+B+A*L
850 F1=1
860 GOSUB 2150
870 GOSUB 3580
880 GOSUB 3360
890 GOSUB 150
900 EG=EG+1
910 IF DF THEN 1030
920 Q=1
930 F1=1
940 GOSUB 2150
950 GOSUB 3580
```

```
960 IF TW<=M*100 THEN 1030
970 GOSUB 7020
980 PRINT @ PL,"carring to much!";
990 GOSUB 6930
1000 GOSUB 2090
1010 GOSUB 6610
1020 GOTO 880
1030 GOSUB 7020
1040 COLOUR 2
1050 PRINT @ PL,"what now SIRE?";
1060 COLOUR 16
1070 A$="IUNSEW"
1080 GOSUB 1940
1090 IF Q=0 THEN 1080
1100 ON Q GOSUB 2090, 2660
1110 IF Q=1 OR Q=2 THEN GOSUB 6610
1120 IF Q=5 AND B<40 THEN B=B+1
1130 IF Q=4 AND A<15 THEN A=A+1
1140 IF Q=6 AND B>0 THEN B=B-1
1150 IF Q=3 AND A>0 THEN A=A-1
1160 IF A<>8 OR O(2, 1)>0 THEN 1230
1170 GOSUB 7020
1180 PRINT @ PL,"YOUR MEN CAN'T SWIM!";
1190 GOSUB 6930
1200 A=A-1
1210 IF Q=3 THEN A=A+2
1220 GOTO 880
1230 F=F+1
1240 P=O(15, 1)-M/2
1250 IF F<10 THEN 1300
1260 F=0
1270 O(15, 1)=P
1280 IF P<M THEN GOSUB 7020: PRINT @ PL,"FOOD LOW!";: GOSUB 6930:IF O(
15, 1)<0 THEN O(15, 1)=0
1290 IF P<M/2 THEN P=RND(M/3): GOSUB 7020: PRINT @ PL, P"men died!";:
M=M-P: GOSUB 6930
1300 MC=MC+1
1310 FOR I=1 TO 32
1320 IF A=D(I, 2)THEN IF B=D(I, 3)THEN POKE N, S1: GOTO 3640
1330 NEXT
1340 POKE N, S1
1350 N=C+A*L+B
1360 IF PEEK(N)<>131 THEN S1=PEEK(N)
1370 DF=0
1380 SWF=0
1390 MF=0
1400 GOSUB 7020
1410 PRINT @ PL,"you are in the ";
1420 B$="wilderness"
1430 IF A=8 THEN B$="river"
1440 IF B>18 THEN IF A<8 THEN B$="mountains": MF=1: F=F+.5
1450 IF S1=130 THEN B$="swamp": SWF=1
1460 IF S1=135 THEN PRINT"shadow of";: GOSUB 6960: PRINT @ PL,"the dre
aded morlock's tower!": GOSUB 6960: GOTO 3390
1470 GOSUB 150
1480 '
1490 FOR J=0 TO 95
1500 GOSUB 150
1510 '
1520 FOR J=0 TO 140
1530 GOSUB 150
```

```
1540
1550 PRINT B$;
1560 POKE N, 131
1570 GOSUB 6770
1580 BT=0
1590 FOR X=1 TO 32
1600 IF ABS(A-D(X, 2))>2 THEN 1710
1610 IF ABS(B-D(X, 3))>2 THEN 1710
1620 IF D(X, 4)=1 THEN D(X, 4)=0
1630 FOR Y=1 TO 9
1640 IF VAL(D$(Y))<>D(X, 1)THEN 1700
1650 IF Y>5 THEN BT=Y
1660 GOSUB 7020
1670 PRINT @ PL, MID$(D$(Y), 4);" near";
1680 GOSUB 6930
1690 GOTO 1710
1700 NEXT Y
1710 NEXT X
1720 IF A>12 THEN IF B>22 THEN GOSUB 7020: PRINT @ PL,"CITADEL near";:
 GOSUB 6930
1730 IF SWF THEN IF RND(100)<50 THEN GOSUB 7020: PRINT @ PL,"2 men are
 stuck in quicksand!";: M=M-2: DF=1: GOSUB 6930: GOTO 1750
1740 IF MF AND S1<>32 THEN IF RND(100)<50 THEN GOSUB 7020: PRINT @ PL,
"2 men fell into a deep ravine!";: M=M-2: DF=1: GOSUB 6930
1750 IF BT THEN 880
1760 BC=BC+1
1770 IF BC<3 THEN 880
1780 BC=0
1790 FOR J=23 TO 32
1800 IF D(J, 4)=0 THEN D(J, 4)=1: POKE C+D(J, 3)+D(J, 2)*L, BQ
1810 X=RND(15)
1820 Y=RND(40)
1830 IF X=8 THEN 1810
1840 FOR ZZ=1 TO 32
1850 IF X=D(ZZ, 2)THEN IF Y=D(ZZ, 3)THEN IF J<>ZZ THEN 1810
1860 NEXT ZZ
1870 D(J, 2)=X
1880 D(J, 3)=Y
1890 NEXT J
1900 IF EX>0 THEN IF RND(100)>80 THEN J=RND(6)+22: D(J, 2)=A: D(J, 3)=
B: GOTO 1300
1910 N=C+B+A*L
1920 POKE N, 131
1930 GOTO 880
1940 B$=INKEY$
1950 Q=0
1960 IF B$="" THEN 1940
1970 FOR I=1 TO LEN(A$)
1980 IF B$=MID$(A$, I, 1)THEN Q=I: RETURN
1990 NEXT
2000 RETURN
2010 A$=""
2020 Q=0
2030 GOSUB 7020
2040 B$=INKEY$
2050 IF B$<>"" THEN IF ASC(B$)=13 THEN Q=VAL(A$): RETURN
2060 IF B$<>"" THEN IF ASC(B$)=8 THEN 2010
2070 IF B$<>"" THEN A$=A$+B$: PRINT @ PL, A$;
2080 GOTO 2040
2090 CLS:COLOUR 2
2100 PRINT TAB(11)"INVENTORY"
```

```
2110 PRINT
2120 PRINT"ITEM            QUANT.  WT. TOTAL"
2130 F1=0
2140 ZL=0
2150 WV=0
2160 TW=0
2170 FOR I=1 TO 17
2180 IF O(I, 1)=0 THEN 2260
2190 J=O(I, 1)
2200 K=O(I, 2)
2210 TW=TW+J*K
2220 IF F1=0 THEN GOSUB 2550: ZL=ZL+1:IF ZL=7 THEN GOSUB 6530: ZL=0
2230 X=O(I, 4)
2240 IF J>M THEN J=M
2250 WV=WV+J*X
2260 NEXT
2270 IF F1 THEN RETURN
2280 PRINT
2290 PRINT
2300 IF Q=3 THEN RETURN
2310 COLOUR 2:PRINT"TOTAL WEIGHT" TW" LBS."
2320 PRINT"your" M" men can carry" M*100
2330 IF Z=5 THEN 6080
2340 PRINT # PL, D$" drop anything?";
2350 FOR W=1 TO 30
2360 NEXT
2370 A$=INKEY$
2380 IF A$="" THEN 2370
2390 IF A$<>"Y" THEN Q=1: RETURN
2400 GOSUB 7020
2410 PRINT # PL, W$;
2420 GOSUB 6960
2430 GOSUB 2010
2440 IF Q>17 THEN 2400
2450 I=Q
2460 QT=O(I, 1)
2470 IF QT=0 THEN GOSUB 7020: PRINT # PL, E$;: GOSUB 6960: GOTO 2340
2480 GOSUB 7020
2490 PRINT # PL, H$;
2500 GOSUB 6960
2510 GOSUB 2010
2520 IF Q>QT THEN PRINT # PL, M$;: GOSUB 6960: GOTO 2340
2530 O(I, 1)=QT-Q
2540 GOTO 2090
2550 COLOUR 5:PRINT MID$(STR$(I), 2)" ";
2560 IF I<10 THEN PRINT" ";
2570 PRINT O$(I); TAB(18)J; TAB(23)MID$(STR$(K), 2); TAB(27)MID$(STR$(
K*J), 2)
2580 RETURN
2590 Z=0
2600 BC=3
2610 BT=0
2620 B=B-1
2630 IF RND(10)>5 THEN B=B+2
2640 N=C+B+A*L
2650 RETURN
2660 CLS
2670 Q=3
2680 GOSUB 2120
2690 GOSUB 7020
2700 PRINT # PL, W$;
```

```
2710 GOSUB 6960
2720 GOSUB 2010
2730 IF Q>17 THEN 2660
2740 I=Q
2750 P=O(I, 1)
2760 IF P=0 THEN GOSUB 7020: PRINT @ PL, E$;: GOSUB 6960: GOTO 2820
2770 IF(I=16 AND Z=1)OR(I=6 AND Z=2)OR(I=4 AND Z=3)OR(I=3 AND Z=4)THEN
 2900
2780 IF I=7 AND DF=1 THEN 3320
2790 GOSUB 7020
2800 PRINT @ PL, O$(I);" - no effect";
2810 GOSUB 6960
2820 GOSUB 7020
2830 PRINT @ PL,"something else?";
2840 GOSUB 6930
2850 A$=INKEY$
2860 IF A$="" THEN 2850
2870 IF A$="Y" THEN 2660
2880 GOSUB 6610
2890 RETURN
2900 GOSUB 6610
2910 GOSUB 7020
2920 PRINT @ PL,"you're in a ";MID$(D$(Z), 4);
2930 GOSUB 6930
2940 J=0
2950 GOSUB 7020
2960 PRINT @ PL,"SEARCH or LEAVE?";
2970 A$=INKEY$
2980 IF A$="" THEN 2970
2990 IF A$="S" THEN J=J+10: GOTO 3020
3000 GOSUB 2590
3010 RETURN
3020 SF=0
3030 IF RND(100)>90 THEN SF=Z: SU=1: Z=9: GOTO 4110
3040 IF RND(100)>J THEN PRINT @ PL,"there's nothing here!";: GOSUB 693
0: GOTO 2950
3050 GOSUB 7020
3060 PRINT @ PL,"there is a CHEST here!";
3070 GOSUB 6930
3080 IF RND(100)>90 THEN 3260
3090 IF D(AL, 4)=-1 THEN GOSUB 7020: PRINT @ PL,"it's empty!";: GOSUB
6930: GOTO 3200
3100 D(AL, 4)=-1
3110 GOSUB 7020
3120 PRINT @ PL,"containing";
3130 IF O(16, 1)OR RND(100)>20 THEN 3170
3140 PRINT" a ";O$(16);
3150 O(16, 1)=1
3160 GOTO 3200
3170 J=RND(500)+50
3180 PRINT J, O$(17);
3190 O(17, 1)=O(17, 1)+J
3200 GOSUB 6960
3210 GOSUB 7020
3220 PRINT @ PL,"Better leave now";
3230 GOSUB 6960
3240 Q=0
3250 GOTO 3000
3260 J=RND(2)
3270 IF M>20 THEN J=RND(5)
3280 GOSUB 7020
3290 PRINT @ PL,"It's a TRAP" J"men are killed";
```

```
3300 M=M-J
3310 GOTO 3200
3320 GOSUB 7020
3330 IF RND(100)<80 THEN PRINT # PL,"The rope broke-it's useless now";
: GOSUB 6960: O(7, 1)=O(7, 1)-1: DF=0 ELSE PRINT # PL,"You saved them!
";: GOSUB 6960: DF=0: M=M+2
3340 GOSUB 6610
3350 RETURN
3360 IF M>0 THEN RETURN
3370 PRINT # PL," ALL YOUR MEN ARE DEAD!"
3380 END
3390 IF TW<2400 THEN GOSUB 7020: PRINT # PL,"Your army is not strong e
nough";: GOSUB 6990: GOSUB 7020: PRINT # PL,"to make succes likely!";:
 GOSUB 6990
3400 IF O(1, 1)=0 THEN GOSUB 7020: PRINT # PL,"The walls are to high";
: GOSUB 6990: PRINT # PL,"for you to besiege the tower!";: GOSUB 6990
3410 GOSUB 7020
3420 PRINT # PL, D$;
3430 GOSUB 6930
3440 PRINT # PL,"Attack or Leave?";
3450 A$="LA"
3460 GOSUB 1940
3470 IF Q=0 THEN 3450
3480 IF Q=2 THEN 3520
3490 IF RND(100)>70 THEN I=RND(M/10)M=M-I: PRINT # PL,"A volley of arr
ows";: GOSUB 6930: GOSUB 7020: PRINT # PL,"from the tower has killed";
: GOSUB 6990: GOSUB 7020: PRINT # PL, I"of your men!";: GOSUB 6990
3500 A=A-1
3510 GOTO 1300
3520 FA=1
3530 BM=RND(40)+160
3540 IW=20
3550 BW=2000
3560 SU=0
3570 GOTO 4500
3580 I=TW/M
3590 FOR SP=1 TO 9
3600 IF 10*SP>=I THEN 3620
3610 NEXT
3620 SP=10-SP
3630 RETURN
3640 AL=I
3650 P=D(I, 1)
3660 FOR Z=1 TO 9
3670 IF P<>VAL(D$(Z)) THEN NEXT: GOTO 880
3680 SU=0
3690 IF Z>5 THEN 4110
3700 GOSUB 7020
3710 PRINT # PL,"you have come to a ";MID$(D$(Z), 4);
3720 GOSUB 6930
3730 ON Z GOTO 3740, 3790, 3850, 3880, 3910
3740 PRINT # PL,"the way is barred by a stout";
3750 GOSUB 6960
3760 PRINT # PL,"door with an iron lock!          ";
3770 GOSUB 6960
3780 GOTO 880
3790 PRINT # PL,"the entrance is filled with";
3800 GOSUB 6960
3810 GOSUB 7020
3820 PRINT # PL,"drifted sand!";
3830 GOSUB 6960
```

```
3840 GOTO 880
3850 PRINT # PL,"it's to dark to see!";
3860 GOSUB 6960
3870 GOTO 880
3880 PRINT # PL,"there is no visible entrance!";
3890 GOSUB 6960
3900 GOTO 880
3910 IF MC<6 THEN 4030
3920 MC=0
3930 FOR I=1 TO 15
3940 O(I, 5)=0
3950 IF RND(100)>50 OR(I=1 AND A>8)THEN IF I<15 THEN 4020
3960 O(I, 5)=RND(20)
3970 V=O(I, 3)
3980 GOSUB 6590
3990 O(I, 6)=V
4000 IF I=1 THEN O(I, 5)=1
4010 IF I=15 THEN O(I, 5)=RND(100)+M
4020 NEXT I
4030 GOSUB 7020
4040 PRINT # PL,"Leave,Buy at market,Inventory?";
4050 A$="LBMI"
4060 GOSUB 1940
4070 IF Q=0 THEN 4030
4080 IF Q=1 THEN GOSUB 2590: GOSUB 6610: GOTO 880
4090 IF Q=4 THEN GOSUB 2090: GOTO 4030
4100 GOTO 5360
4110 GOSUB 150
4120 '
4130 GOSUB 7120
4140 '
4150 GOSUB 7020
4160 PRINT # PL,"You have encountered a band of";
4170 GOSUB 6930
4180 GOSUB 7020
4190 PRINT # PL, MID$(D$(Z), 4);
4200 GOSUB 6930
4210 BM=RND(M+M/4)+INT(EG/10)
4220 BS=RND(10)
4230 BX=RND(INT(EG/10))
4240 SB=RND(10)
4250 BI=BM
4260 BX=INT(BX)
4270 IF Z<8 THEN GOSUB 7020: PRINT # PL,"they have an ESCORT!";: GOSUB
 6930
4280 IW=RND(8)-10*(EX>2)
4290 BW=BM*IW
4300 GOSUB 7020
4310 PRINT # PL,"There are" BM".";
4320 IF IW<6 THEN PRINT" LIGHTLY"; ELSE IF I>5 AND IW<15 THEN PRINT" W
ELL"; ELSE PRINT"HEAVILY";
4330 PRINT" armed";
4340 GOSUB 6930
4350 IF SU THEN 4470
4360 GOSUB  7020
4370 PRINT # PL, D$;
4380 GOSUB  6930
4390 PRINT # PL,"Attack,Run or be Friendly?";
4400 GOSUB  6930
4410 A$="ARF"
4420 GOSUB  1940
```

```
4430 ON Q GOTO 4450. 4860. 5020
4440 GOTO 4360
4450 SU=-1
4460 IF RND(100)>.50 THEN SU=0: GOTO 4500
4470 GOSUB  7020
4480 B$="they"
4490 IF SU=1 THEN B$="you": PRINT @ PL. B$" were surprised";: GOSUB  6
930
4500 F1=1
4510 GOSUB  2150
4520 BW=BM*IW
4530 CLS
4540 PRINT TAB(12)"You";TAB(22)"Them"
4550 PRINT"Men" TAB(12)M; TAB(22)BM
4560 PRINT"Weapons";TAB(12)WV; TAB(22)BW
4570 PRINT"Experience";TAB(12)EX; TAB(22)BX
4580 PRINT @ 400,"LUCK";
4590 GOSUB 150
4600 '
4610 I=RND(100)
4620 J=RND(100)
4630 PRINT @ 415. I"        " J;
4640 A$=INKEY$
4650 IF A$="" THEN 4590
4660 I=I-SU*10
4670 J=J+SU*10
4680 TW=WV+I+ST*5+EX*5
4690 IF FA THEN IF O(1, 1)=0 THEN TW=TW/5
4700 WT=BW+J+SB*2+BX*2
4710 M=M+INT((WT>TW)*RND(M/10)+(TW>WT)*( RND(M/20)-1))
4720 BM=BM+INT((TW>WT)*RND(BW/10)+(WT>TW)*( RND(BM/20)-1))
4730 IF BM<0 THEN BM=0
4740 GOSUB  3360
4750 PRINT
4760 PRINT"Survivors";TAB(12)M; TAB(22)BM
4770 IF BM=0 THEN 5240
4780 IF BM<BI/4 AND RND(100)>50 AND FA=0 THEN 5050
4790 PRINT
4800 PRINT"Fight, Run or Terms?"
4810 A$="FRT"
4820 GOSUB  1940
4830 IF Q=0 THEN 4790
4840 IF Q=2 THEN 4860
4850 IF Q=3 THEN 4920 ELSE 4500
4860 IF SB>B OR RND(100)>90 OR FA=1 THEN GOSUB  7020: PRINT @ PL,"You
got away!";: GOSUB  6930: FA=0: GOTO 5150
4870 GOSUB  7020
4880 PRINT @ PL,"They caught you!";
4890 GOSUB  6930
4900 SU=1
4910 GOTO 4470
4920 I=WT*RND(100)
4930 J=TW*RND(10)*CH
4940 GOSUB  6930
4950 IF J>I AND FA=0 THEN 5000
4960 PRINT @ PL, A$;
4970 GOSUB  6930
4980 SU=1
4990 GOTO 4470
5000 BM=RND(BM)
5010 GOTO 5050
5020 GOSUB  7020
```

```
5030 IF RND(10)>CH THEN PRINT $ PL, R$;: GOSUB  6960: SU=1: GOTO 4470
5040 BM=RND(BM)
5050 PRINT $ PL, BM"will join you!";
5060 GOSUB  6930
5070 M=M+BM
5080 PRINT
5090 PRINT"They have:"
5100 PRINT
5110 FOR I=8 TO 15
5120 IF RND(100)>50 THEN QT=RND(BM): O(I, 1)=O(I, 1)+QT: PRINT QT: O$(
I)
5130 NEXT
5140 IF RND(100)>90 THEN PRINT"Too bad they had no GOLD!" ELSE QT=INT(
RND(5)*((EG/10)+1)*BM): O(17, 1)=O(17, 1)+QT: PRINT QT: O$(17)
5150 EX=EX+1
5160 PRINT
5170 PRINT
5180 PRINT $ PL,"READY TO GO ON?";
5190 A$="A"
5200 GOSUB  1940
5210 IF SF THEN Z=SF: GOTO 2900
5220 GOSUB  6610
5230 GOTO 1780
5240 IF FA THEN 6820
5250 FOR J=1 TO 2
5260 GOSUB 150
5270 '
5280 GOSUB  7120
5290 '
5300 GOSUB 7120
5310 NEXT
5320 PRINT"YOU WON!
5330 BM=2
5340 PRINT"You managed to salvage:"
5350 GOTO 5110
5360 HC=0
5370 OC=0
5380 ZL=0
5390 CLS
5400 VF=1
5410 PRINT"ITEM                    PRICE   WT.  QUANT."
5420 PRINT
5430 FOR I=1 TO 15:COLOUR 3
5440 IF O(I, 5)=0 THEN 5500
5450 PRINT MID$(STR$(I), 2)" ";
5460 IF I<10 THEN PRINT" ";
5470 PRINT O$(I); TAB(18)O(I, 6); TAB(23)O(I, 2); TAB(27)O(I, 5)
5480 ZL=ZL+1
5490 IF ZL=7 THEN GOSUB 6480:IF ZL=99 THEN 5510
5500 NEXT I
5510 HB=0
5520 F1=1
5530 GOSUB 2150
5540 PRINT"You have" O(17, 1)"Gold Coins"
5550 PRINT"Your men are carrying" TW"lbs"
5560 GOSUB 7020
5570 PRINT $ PL, W$;
5580 GOSUB 6960
5590 GOSUB 2010
5600 IF Q>17 THEN 5560
5610 V=O(Q, 6)
```

```
5620 OL=3+RND(4)-(5*(V>150))
5630 GOSUB 6590
5640 X=Q
5650 GOSUB 7020
5660 IF O(X, 5)=0 THEN PRINT $ PL, E$;: GOSUB 6960: GOTO 4030
5670 GOSUB 7020
5680 PRINT $ PL,"Your offer?";
5690 GOSUB 6960
5700 GOSUB 2010
5710 Q=INT(Q)
5720 GOSUB 7020
5730 IF Q>=V THEN V=Q: GOTO 5930
5740 IF RND(30)=15 THEN V=Q: GOTO 5930
5750 IF RND(20)=12 THEN OL=OL+10
5760 IF RND(100)=74 THEN V=V*2
5770 IF Q<V/3 THEN PRINT $ PL,"not interested!";: GOTO 6050
5780 IF Q=HB THEN HC=HC+1
5790 IF Q<HB OR(Q=HB AND HC=2)THEN 6030
5800 HB=Q
5810 PRINT $ PL,"Tell you what I'm going to do--";
5820 GOSUB 6930
5830 ZZ=INT(V*((RND(15)+5)*.01))
5840 IF ZZ<1 THEN ZZ=1
5850 V=V-ZZ
5860 IF V<HB THEN V=HB
5870 IF Q>=V THEN V=Q: GOTO 5930
5880 GOSUB 7020
5890 IF RND(100)>50 THEN PRINT $ PL,"For you I'll make it" V"OK?"; ELS
E PRINT $ PL,"What do you say to" V"?";
5900 A$=INKEY$
5910 IF A$="" THEN 5900
5920 IF A$<>"Y" THEN OC=OC+1:IF OC<OL THEN 5670 ELSE 6030
5930 GOSUB 7020
5940 PRINT $ PL, H$;
5950 GOSUB 6960
5960 GOSUB 2010
5970 IF Q>O(X, 5)THEN PRINT $ PL, M$;: GOSUB 6960: GOTO 5930
5980 IF Q*V>O(17, 1)THEN GOSUB 7020: PRINT $ PL,"not enough GOLD!";: G
OSUB 6960: GOTO 5930
5990 O(X, 1)=O(X, 1)+Q
6000 O(17, 1)=O(17, 1)-Q*V
6010 O(X, 5)=O(X, 5)-Q
6020 GOTO 4030
6030 GOSUB 7020
6040 PRINT $ PL, Y$;
6050 GOSUB 6960
6060 O(X, 6)=INT(O(X, 6)/4)+O(X, 6)
6070 GOTO 4030
6080 PRINT $ PL, D$" sell anything?";
6090 A$=INKEY$
6100 IF A$="" THEN 6080
6110 IF A$<>"Y" THEN RETURN
6120 GOSUB 7020
6130 PRINT $ PL, W$;
6140 GOSUB 6960
6150 GOSUB 2010
6160 IF Q>17 THEN 6120
6170 I=Q
6180 IF O(I, 1)=0 THEN PRINT $ PL, E$;: GOSUB 6960: GOTO 6080
6190 V=INT(O(I, 3)/( RND(3)+2))
6200 OL=5+RND(5)-10+(V>80)
6210 HB=O(I, 3)+1
```

```
6220 HC=0
6230 IF RND(100)>50 THEN PRINT # PL,"I'll give you" V"OK?": ELSE PRINT
 # PL,"Will you take" V"?":
6240 A$=INKEY$
6250 IF A$="" THEN 6240
6260 IF A$="Y" THEN 6400
6270 GOSUB 7020
6280 PRINT # PL,"How much do you want?";
6290 GOSUB 6960
6300 GOSUB 2010
6310 J=Q
6320 IF J=HB THEN HC=HC+1 ELSE HC=0
6330 IF J<=V THEN 6400
6340 IF J>HB OR(J=HB AND HC=2)OR OL=0 THEN PRINT # PL, Y$:: GOSUB 6960
: RETURN
6350 HB=J
6360 ZZ=INT(V*( RND(20)+10)*.01)
6370 IF ZZ<1 THEN ZZ=1
6380 V=V+ZZ
6390 IF V<J THEN 6230
6400 GOSUB 7020
6410 PRINT # PL, H$;
6420 GOSUB 6960
6430 GOSUB 2010
6440 IF Q>O(I, 1)THEN PRINT # PL, M$:: GOSUB 6960: GOTO 6400
6450 O(17, 1)=O(17, 1)+V*Q
6460 O(I, 1)=O(I, 1)-Q
6470 RETURN
6480 J=0
6490 GOTO 6580 PRINT"press <RETURN> to continue, or"
6500 PRINT"<SPACE> to buy!"
6510 PRINT
6520 GOTO 6550
6530 RETURN
6540 PRINT"press <RETURN> to continue"
6550 A$=INKEY$
6560 IF A$="" THEN 6550
6570 IF ASC(A$)=32 THEN ZL=99
6580 RETURN
6590 V=V-RND((V/4))
6600 RETURN
6610 CLS
6620 FOR I=0 TO 10
6630 FOR J=0 TO 3
6640 IF RND(10)>5 THEN POKE SL+I+J*L, 130 : POKE SL+I+J*L-21504,6
6650 NEXT J, I
6660 FOR I=0 TO 38 STEP 2
6670 POKE 17728+I, 132 : POKE I-3776, 8
6680 POKE 17729+I, 133 : POKE I-3775, 8
6690 NEXT
6700 FOR I=1 TO 21 STEP 2
6710 FOR J=0 TO 6
6720 IF RND(10)>6 THEN POKE 17426+I+J*L, 128: POKE 17427+I+J*L, 129: P
OKE I+J*L-4078,15 : POKE I+J*L-4077,15
6730 NEXT J, I
6740 POKE 17999, 134: POKE-3505,10
6750 POKE 18000, 135: POKE-3504,10
6760 POKE 18001, 136: POKE-3503,10
6770 POKE C+B+A*L, 131
6780 FOR X=1 TO 32
6790 IF D(X, 4)<1 THEN POKE C+D(X, 2)*L+D(X, 3), D(X, 1) : POKE C+D(X,
```

```
2)*L+D(X,3)-21504,2
6800 NEXT
6810 RETURN
6820 CLS
6830 FOR I=1 TO 5
6840 GOSUB 150
6850 SOUND 4,89
6860 GOSUB 7120
6870 GOSUB 150
6880 SOUND 4,176
6890 GOSUB 7120
6900 NEXT
6910 COLOUR5:PRINT $ 400,"      YOU'VE CONQUERED THE":PRINT"
FORCES":PRINT"         OF THE EVIL!":PRINT"   ALL HAIL THE VICTOR!!!"
6920 GOTO 6820
6930 FOR WT=1 TO 300
6940 NEXT
6950 RETURN
6960 FOR WT=0 TO 600
6970 NEXT
6980 RETURN
6990 FOR WT=1 TO 1000
7000 NEXT
7010 RETURN
7020 PRINT $ PL, STRING$(32, 32):
7030 RETURN
7040 PRINT $ 409,"RUN TO MORLOCK'S TOWER";
7050 COLOUR 16
7060 PRINT $ 454,"The ultimate";
7070 PRINT $ 493,"Adventure Game";
7080 COLOUR 9
7090 PRINT $ 850,"Press <RETURN> to play";
7100 COLOUR 16
7110 RETURN
7120 FOR WT=1 TO 25
7130 NEXT
7140 RETURN
7150 CLS
7160 COLOUR 5
7170 PRINT"EINLEITUNG:"
7180 COLOUR 16
7190 PRINT"IN DIESEM ADVENTURESPIEL SIND SIE DER   FUEHRER EINER RAUBR
ITTERBANDE UND        STREIFEN DURCHS LAND, UM SICH GELD,     WAFFEN UN
D NEUE SOLDATEN ZU BESCHAFFEN, UM SCHLIESSLICH IN DEN < MORLOCK'S "
7200 PRINT"TOWER > EINZUDRINGEN. DORT MUESSEN SIE  DANN GEGEN DIE KRAE
FTE DES BOESEN        KAEMPFEN UND DIESE BESIEGEN, UM SELBST  HERR UEBE
R DAS LAND ZU WERDEN. DAS SPIELIST IN ENGLISCHER SPRACHE GESCHRIEBEN,"
7210 PRINT"DA DIESE KUERZER UND PRAEZISER ALS DIE  DEUTSCHE IST."
7220 PRINT"WENN MAN DEM SPIEL AUFMERKSAM FOLGT,    WERDEN WOHL KEINE P
ROBLEME AUFTRETEN."
7230 PRINT
7240 COLOUR 9
7250 PRINT"<RETURN>"
7260 COLOUR 5
7270 A$=INKEY$
7280 IF A$="" THEN 7270
7290 GOTO 7300
7300 CLS
7310 PRINT"LANDKARTE:"
7320 COLOUR 16
7330 PRINT"DIE LANDKARTE IST WAEHREND DER MEISTEN  ZEIT SICHTBAR, SIE
ZEIGT AN, WO SICH DERSPIELER, MIT DEM KLEINEN MAENNCHEN      SYMBOLISI
```

```
ERT. GERADE BEFINDET UND WO    DIE VERSCHIEDENEN INTERESSANTEN PUNKTE
"
7340 PRINT"LOKALISIERT SIND. MAN KANN DIE SPIEL-    FIGUR MIT DEN TASTE
N FUER DIE VIER       HIMMELSRICHTUNGEN, ALSO ";
7350 COLOUR 3
7360 PRINT"N";
7370 COLOUR 16
7380 PRINT")ORTH,";
7390 COLOUR 3
7400 PRINT"S";
7410 COLOUR 16
7420 PRINT")OUTH, ";
7430 COLOUR 3
7440 PRINT" W";
7450 COLOUR 16
7460 PRINT")EST UND ";
7470 COLOUR 3
7480 PRINT"E";
7490 COLOUR 16
7500 PRINT")AST BEWEGEN"
7510 PRINT"FOLGENDE OBJEKTE WERDEN DARGESTELLT:    DA IST ZUM EINEN EI
NE GEWELLTE LINIE,   DIE SICH QUER DURCH DIE KARTE ERSTRECKT.SIE STELL
T EINEN FLUSS DAR, DER NUR      MITTELS EINES BOOTES DURCHQUERT WERDEN
 KANN. DIE GEPUNKTETEN FLAECHEN "
7520 PRINT"STELLEN EINEN SUMPF DAR (SWAMP), IN DEM IHRE SOLDATEN EINSI
NKEN KOENNEN, DIE     DANN NUR MITTELS EINES SEILS (ROPE)      GERETTET
WERDEN KOENNEN. IM NORDOEST-"
7530 PRINT"LICHEN TEIL DES GEBIETES BEFINDET SICH  EIN GEBIRGE, DAS TE
ILWEISE MIT STILI   SIERTEN BERGEN ANGEDEUTET IST."
7540 COLOUR 9
7550 PRINT"     <RETURN>"
7560 COLOUR 16
7570 A$=INKEY$
7580 IF A$="" THEN 7570
7590 CLS
7600 PRINT"GANZ IM SUEDOSTEN IST EINE BURGANLAGE   SICHTBAR. DORT IST"
7610 COLOUR 3
7620 PRINT"<MORLOCK'S TOWER>";
7630 COLOUR 16
7640 PRINT", DAS ENDZIEL IHRES ABENTEUERS. AUF DER KARTE SIND AUSSERDE
M   ZWEI 'k's. DIESE BEZEICHNEN DEN STAND-"
7650 PRINT"ORT VON STAEDTEN, WO SIE VERSCHIEDENE "
7660 PRINT"DINGE EINKAUFEN KOENNEN. BEIM DURCH-    WANDERN DER LANDSCH
AFT ERSCHEINEN IN     IHRER NAEHE MANCHMAL NOCH ANDERE          ZEICHEN,
DIESE HABEN DIE SYMBOLIK:"
7670 PRINT"C = CASLTLE (BURG)                           c = CAVE (HOEHLE OD
ER KAVERNE)           R = RUIN (RUINE)                             T = TOWER
 (TURM)                        S = SOLDIERS (SOLDATEN)
 O = OGRES (ORGS)           "
7680 PRINT"P = PILGRIMS (PILGER)                        W = WIZZARDS (GELEH
RTE)
7690 COLOUR 9
7700 PRINT
7710 PRINT
7720 PRINT
7730 PRINT"<RETURN>"
7740 COLOUR 5
7750 A$=INKEY$
7760 IF A$="" THEN 7750
7770 CLS
7780 PRINT"STADTBESUCH:"
7790 COLOUR 16
```

```
7800 PRINT"BRINGT MAN SEINE FIGUR MIT EINER STADT  (CITY) DECKUNGSGLEI
CH, DANN KANN MAN    DORT AUF DEN MARKTPLATZ GEHEN. DAS ANGE-BOT IST N
ICHT IMMER VOLLSTAENDIG, UND    MAN SOLLTE AUF DIE ARTIKEL ACHTEN. "
7810 PRINT"MAN BRAUCHT STEIGEISEN (GRABBLING HOOKS)UM IN EINEN TURM EI
NDRINGEN ZU KOENNEN, GENAUSO WIE OELLAMPEN (OILLAMPS) UM IN  HOEHLEN E
INZUDRINGEN. EIN BOOT WAERE    NOETIG UM DEN FLUSS ZU UEBERQUEREN. UM
 IN EINE RUINE EINZUDRINGEN SIND "
7820 PRINT"SCHAUFELN (SHOVELS) NOETIG UND IN EINE  BURG KOMMT MAN NUR
MIT EINEM GROSSEN    SCHLUESSEL, DEN MAN ABER NUR IM SCHATZ  EINES TUR
MS, EINER RUINE ODER EINER     HOEHLE FINDEN KANN. AUCH IST ES SINNVOL
LNAHRUNGSPAKETE (RATIONS) ZU KAUFEN. "
7830 PRINT"UM DIE MANNSCHAFT SATT ZU HALTEN.       DIE PREISE KANN MAN
 MIT IMAGINAEREN     HAENDLER AUSHANDELN, INDEM MAN SICH AN  DEM GEFOR
DERTEN PREIS ORIENTIERT."
7840 COLOUR 9
7850 PRINT"<RETURN>"
7860 COLOUR 16
7870 A$=INKEY$
7880 IF A$="" THEN 7870
7890 CLS
7900 COLOUR 5
7910 PRINT"INVENTORY:"
7920 COLOUR 16
7930 PRINT"DAS INVENTORY IST EINE UEBERSICHT UEBER ALLE SACHEN, DIE MA
N BESITZT, SOWIE DIE ANZAHL DER EIGENEN SOLDATEN UND DEREN   TRAGKRAFT
. WAEHLT MAN DAS INVENTORY IN"
7940 PRINT"DET R WILDNIS,  SO KANN MAN UEBER ES      ARTIKEL WEGWERFEN
, WAS NOETIG WIRD, WENNDIE SOLDATEN DIE GESAMTE LADUNG NICHT   MEHR TR
AGEN KOENNEN."
7950 COLOUR 5
7960 PRINT"EINDRINGEN:"
7970 COLOUR 16
7980 PRINT"WILL MAN IN EINE BURG, EINEN TURM, EINE RUINE ODER IN EINE
HOEHLE EINDRINGEN, SOMUSS MAN SEINE FIGUR DAZU DECKUNGSGLEICHBRINGEN U
ND 'U' FUER 'USE' BENUTZEN     DRUECKEN. DANN WIRD DAS INVENTORY AUF-
 GELISTET ";
7990 PRINT"UND NACH DEN ZU BENUTZENDEN    ARTIKEL GEFRAGT. IST DIESER
RICHTIG,    SO WIRD DIE KARTE GEZEICHNET UND MAN    WIRD GEFRAGT, OB M
AN SUCHEN ODER ODER   DEN ORT VERLASSEN WILL. NACH ERFOLG-    REICHER
SUCHE KANN MAN ENTWEDER EINEN   SCHATZ (CHEST) ";
8000 PRINT"BZW. FEINLICHE SOLDATEN  FINDEN"
8010 COLOUR 9
8020 PRINT"<RETURN>"
8030 COLOUR 5
8040 A$=INKEY$
8050 IF A$="" THEN 8040
8060 CLS
8070 PRINT"KAMPF:"
8080 COLOUR 16
8090 PRINT"MAN KANN MIT SOLDATEN ODER DER ESKORTE  VON PILGERN ODER GE
LEHRTEN KAEMPFEN.    BEGEGNET MAN SOLCHEN GIBT DER RECHNER   DIE ANZAH
L UND DIE KAMPFSTAERKE (LIGHT, WELL ODER HEAVY) DER GEGNER AN. MAN "
8100 PRINT"HAT DANN DIE WAHL MIT IHNEN ZU KAEMPFEN (ATTACK BZW. FIGHT)
, WEGZULAUFEN(RUN)   ODER SICH MIT INHEN ANZUFREUNDEN       (FRIENDLY
, BZW TERMS). DER KAMPF IST BE-ENDET, WENN SIE ODER DER GEGNER VER-
 NICHTET SIND,  SIE GEFLUECHTET SIND "
8110 PRINT"ODER FEINDE ZU IHNEN UEBERGELAUFEN SIND.".
8120 PRINT"<RETURN>"
8130 A$=INKEY$
8140 IF A$="" THEN 8130
8150 GOTO 80
```

```
8160 POKE -3071, 32
8170 POKE -3070, 112
8180 POKE -3069, 83
8190 POKE -3068, 122
8200 POKE -3067, 255
8210 POKE -3066, 203
8220 POKE -3065, 254
8230 POKE -3064, 64
8240 POKE -3063, 224
8250 POKE -3062, 248
8260 POKE -3061, 184
8270 POKE -3060, 212
8280 POKE -3059, 222
8290 POKE -3058, 187
8300 POKE -3057, 255
8310 POKE -3056, 9
8320 POKE -3055, 72
8330 POKE -3053, 81
8340 POKE -3052, 4
8350 POKE -3051, 73
8360 POKE -3049, 41
8370 POKE -3048, 60
8380 POKE -3047, 188
8390 POKE -3046, 152
8400 POKE -3045, 255
8410 POKE -3044, 61
8420 POKE -3043, 37
8430 POKE -3042, 36
8440 POKE -3041, 36
8450 POKE -3038, 128
8460 POKE -3037, 193
8470 POKE -3036, 227
8480 POKE -3035, 127
8490 POKE -3034, 62
8500 POKE -3033, 28
8510 POKE -3031, 60
8520 POKE -3030, 126
8530 POKE -3029, 255
8540 POKE -3028, 195
8550 POKE -3027, 129
8560 POKE -3023, 146
8570 POKE -3022, 254
8580 POKE -3021, 254
8590 POKE -3020, 84
8600 POKE -3019, 127
8610 POKE -3018, 93
8620 POKE -3017, 95
8630 POKE -3016, 18
8640 POKE -3015, 31
8650 POKE -3014, 31
8660 POKE -3013, 31
8670 POKE -3012, 30
8680 POKE -3011, 255
8690 POKE -3010, 171
8700 POKE -3009, 255
8710 POKE -3008, 73
8720 POKE -3007, 255
8730 POKE -3006, 255
8740 POKE -3005, 255
8750 POKE -3004, 103
8760 POKE -3003, 255
8770 POKE -3002, 249
```

```
8780 POKE -3001, 249
8790 RETURN
8800 COLOUR 14
8810 PRINTCHR$(205);STRING$(38,202);CHR$(224);
8820 FOR W=1 TO 21
8830 PRINTCHR$(202);STRING$(38,32);CHR$(202);
8840 NEXT
8850 PRINTCHR$(228);STRING$(38,202);CHR$(226);
8860 COLOUR 15
8870 RETURN
```

BAR

Gehören Sie auch zu den Leuten, die meinen, man müsse für seinen Computer möglichst viele sinnvolle Anwendungen finden? Nun hier ist eine: Unser Colour-Genie-Programm "Bar" !
Aus fast 80 verschiedenen Drinks-Rezepten sucht das Programm sicher einen Drink heraus, der Ihnen schmeckt. Dabei kann man alle Rezepte ausgeben, bestimmte Drinks nach ihrem Namen suchen, oder als Clou eine Liste der verfügbaren Zutaten eingeben, woraufhin das Colour-Genie alle Rezepte ausgibt, die mit den eingegebenen Zutaten gemixt werden können.

Noch zwei Hinweise zum Eintippen:

- Wir haben das Programm mit 40 Zeichen pro Zeile gelistet, damit Sie mit der Bildschirmformatierung keine Schwierigkeiten haben.

- Lassen Sie sich nicht schon beim Eintippen zum Probieren der Rezepte verleiten, sonst riskieren Sie u.U. etliche Tippfehler.

(Für dieses Programm muß Ihr Colour-Genie 32K RAM haben.)

Programmlisting:

```
10 POKE 16553, 255
20 CLEAR 600
30 DEFINT I, J
40 DEFSTR D, N, M, R, V, X, S, L
50 DIM R(11), M(11), V(50), L(50)
60 ON ERROR GOTO 2860
70 CLS
80 FOR I=1 TO 239
90 COLOUR RND(16)
100 PRINT"BAR ";
110 NEXT
120 PRINT"BAR";
130 FOR I=1 TO 500
140 NEXT I
150 COLOUR 2
160 CLS
170 PRINT @ 405,"DRINKS,DRINKS AND MORE
DRINKS !"
180 FOR I=1 TO 800
190 NEXT I
200 CLS
210 COLOUR 2
220 PRINT @ 5,"*========*** DRINKS ***=====
==*"
230 PRINT
240 COLOUR 5
250 PRINT"WOLLEN SIE"
260 COLOUR 5
```

```
270 PRINT
280 PRINT"1) AUFLISTUNG ALLER VORHANDENE
N REZEPTE"
290 PRINT"2) WISSEN WAS SIE MIT IHREN ZU
TATEN        MIXEN KOENNEN"
300 PRINT"3) DAS REZEPT FUER EINEN GANZ
BESTIMMTEN   DRINK
310 COLOUR 2
320 PRINT
330 PRINT"FUER DIE RUECKKEHR ZU DIESER L
ISTE BITTE ";
340 COLOUR 3
350 PRINT"'R' ";
360 COLOUR 2
370 PRINT"DRUECKEN!
380 D=INKEY$
390 IF D="" THEN 380
400 ID=VAL(D)
410 ON ID GOTO 420, 680, 1740
420 COLOUR 7
430 PRINT
440 PRINT"ZUM ANHALTEN DER AUFLISTUNG BI
TTE         'SHIFT & @'GLEICHZEITIG DRUECK
EN"
450 FOR I=1 TO 200
460 NEXT
470 RESTORE
480 READ N
490 PRINT
500 COLOUR 16
510 PRINT STRING$(40, 217);
520 PRINT N
530 COLOUR 5
540 FOR I=1 TO 400
550 NEXT
560 READ M
570 PRINT
580 PRINT M
590 PRINT
600 I=0
610 RE=INKEY$
620 IF RE="R" THEN RE="":GOTO 200
630 COLOUR 3
640 READ M(I), R(I)
650 IF LEFT$(R(I), 1)="*" THEN: N=R(I):
GOTO 490
660 PRINT M(I), R(I)
670 GOTO 630
680 PRINT
690 COLOUR 14
700 PRINT"BITTE TIPPEN SIE JEWEILS J (=J
A) FUER   JEDE ZUTAT EIN, DIE SIE VORRAE
TIG HABEN.ANDERNFALLS 'N' EINTIPPEN."
710 FOR I=0 TO 50
720 V(I)="N"
730 NEXT I
740 I=0
```

```
750 L(0)="ZUCKER"
760 L(1)="ZITRONE"
770 L(2)="ORANGE"
780 L(3)="ANANAS"
790 L(4)="GRAPEFRUIT"
800 L(42)="KIRSCHE"
810 L(5)="STAUBZUCKER"
820 L(6)="ANGOSTURA"
830 L(7)="GRENADINE"
840 L(8)="WERMUT"
850 L(9)="GIN"
860 L(10)="RUM"
870 L(11)="WHISKY"
880 L(12)="SEKT"
890 L(13)="CURACAO"
900 L(14)="GRAND MARNIER"
910 L(15)="COINTREAU"
920 L(16)="APRICOT BRANDY"
930 L(17)="CHERRY BRANDY"
940 L(18)="PRINZEN KIRSCH"
950 L(19)="MARASCHINO"
960 L(20)="BANANENLIKOER"
970 L(21)="KOKOSNUSSCREME"
980 L(22)="PFEFFERMINZLIKOER"
990 L(23)="CREME DE KAKAO"
1000 L(24)="KAFFEELIKOER"
1010 L(25)="CALVADOS"
1020 L(26)="CORDIAL MEDOC"
1030 L(27)="SOUTHERN COMFORT"
1040 L(28)="WODKA"
1050 L(29)="COGNAC"
1060 L(30)="BRANDY"
1070 L(31)="PORT"
1080 L(32)="SHERRY"
1090 L(33)="ROTWEIN"
1100 L(34)="TONIC WASSER"
1110 L(35)="GINGER ALE"
1120 L(36)="ROSES LIME JUICE"
1130 L(37)="TRINKSCHOKOLADE"
1140 L(38)="MOCCA"
1150 L(39)="EI"
1160 L(40)="ZITRONENSORBET"
1170 L(41)="VANILLE EIS"
1180 L(42)="TEQUILA"
1190 L(43)="CASSIS"
1200 L(44)="BENEDICTINE"
1210 L(45)="CAMPARI"
1220 PRINT
1230 FOR J=0 TO 45
1240 COLOUR(J AND 1)+2
1250 PRINT L(J);
1260 PRINT TAB(20); CHR$(14);
1270 X=INKEY$
1280 IF X="" THEN 1270
1290 IF X="J" THEN V(I)=L(J): I=I+1
1300 PRINT X
```

```
1310 NEXT J
1320 PRINT CHR$(15);
1330 PRINT
1340 PRINT"SUCHE BEGINNT - BITTE WARTEN"
1350 RESTORE
1360 READ N
1370 READ M
1380 I=0
1390 READ M(I), R(I)
1400 IF LEFT$(R(I), 1)="*" THEN: NA=R(I)
: R(I)="": GOTO 1430
1410 I=I+1
1420 GOTO 1390
1430 I=0
1440 J=0
1450 RE=INKEY$
1460 IF RE="R" THEN: RE="": GOTO 200
1470 IF LEFT$(R(I), LEN(V(J)))=V(J)THEN
1550
1480 J=J+1
1490 IF V(J)<>"N" THEN 1470
1500 FOR I=0 TO 11
1510 R(I)=""
1520 NEXT I
1530 N=NA
1540 GOTO 1370
1550 I=I+1
1560 IF R(I)<>"" THEN 1440
1570 PRINT
1580 COLOUR 16
1590 PRINT STRING$(40, 217)
1600 COLOUR 5
1610 PRINT N
1620 PRINT
1630 COLOUR 2
1640 PRINT M
1650 PRINT
1660 I=0
1670 COLOUR 3
1680 PRINT M(I), R(I)
1690 I=I+1
1700 IF R(I)<>"" THEN 1670
1710 FOR I=1 TO 999
1720 NEXT I
1730 GOTO 1500
1740 PRINT
1750 INPUT"WELCHEN DRINK SUCHEN SIE";S
1760 IF RIGHT$(S, 1)=" " THEN: S=LEFT$(S
, LEN(S)-1): GOTO 1760
1770 S="* " +S
1780 RESTORE
1790 FOR I=1 TO LEN(S)
1800 IF MID$(S, I, 1)=" " THEN IF MID$(S
, I+1, 1)=" " THEN: S=LEFT$(S, I)+RIGHT$
(S, LEN(S)-I-1): GOTO 1790
```

```
1810 NEXT
1820 PRINT S
1830 READ D
1840 RE=INKEY$
1850 IF RE="A" THEN: RE="": 40
1860 IF LEFT$(D, LEN(S))<>S THEN 1830
1870 N=D
1880 PRINT
1890 PRINT N
1900 I=0
1910 READ M
1920 PRINT
1930 COLOUR 16
1940 PRINT STRING$(40,217)
1950 PRINT M
1960 PRINT
1970 READ M(I), A(I)
1980 IF LEFT$(A(I), 1)="*" THEN 2040
1990 COLOUR5
2000 PRINT M(I),
2010 COLOUR2
2020 PRINT A(I)
2030 GOTO 1970
2040 RE=INKEY$
2050 IF RE<>"A" THEN: RE="": GOTO 2040 E
LSE 200
2060 DATA* RAUHREIF-COCKTAIL, SHAKER MIT
 EIS - ZUTATEN - GUT          SCHUETTELN
 UND IN EIN COCKTAIL-  GLAS   MIT ZUCKER
RAND SEIHEN, 1/3, GIN, 1/3, CURACAO TRIP
LE SEC, 1/3, ZITRONENSAFT, 1 SPRITZER, R
UM
2070 DATA 0,* LONG LUCAS, MIXBECHER MIT
EIS - ZUTATEN - MIT SEKT  AUFFUELLEN - 1
 ORANGEN - SCHEIBE          EINLEGEN - MIT
 ZITRONENSCHEIBE GARNIEREN, 1/2, GIN, 1/
2, CURACAO ORANGE, SAFT 1, ORANGE, 1/2,
ZITRONE
2080 DATA 0,* INFERNO, IN EINER SEKTSCHA
LE ZUTATEN MIT ZITRONENSCHALE ABSPRITZEN
 - DANN MIT EINEM        SCHUSS FRISCHEM O
RANGENSAFT UND SEKT AUFFUELLEN -    EISS
TUECKE BEIGEBEN, 1/3, GIN, 2/3, CURACAO
ORANGE, , ORANGENSAFT, , SEKT
2090 DATA 0,* GIN FIZZ, IM SHAKER MIT KL
EINGESTOSSENEM EIS       ZUTATEN KRAEFTIG
 SCHUETTELN  IN EIN GLASGEBEN UND MIT SO
DA AUFSPRITZEN, 2/3, GIN, 1/3, ZITRONENS
AFT, 1 TL, STAUBZUCKER
2100 DATA 0,* SILVER MOON, ZUTATEN MIT E
IS MIXEN UND EINER HALBEN  SCHEIBE ORANG
E GARNIEREN, 1/3, GIN, 1/3, APRICOT BRAN
DY, 1/3, ORANGENSAFT
2110 DATA 0,* MARTINI DRY, ZUTATEN MIT E
ISWUERFELN IM MIXBECHER GUTVERRUEHREN: D
URCHSEIHEN - IM          COCKTAILGLAS
MIT OLIVE SERVIEREN - NACH WUNSCH MIT ZI
```

```
TRONENSCHALE ABSPRITZEN, 2 DASH, ORANGE-
BITTER, 1/2, WERMUT DRY, 1/2, GIN
2120 DATA 0,* WHITE LADY, SHAKER MIT EIS
 - ZUTATEN - GUT           SCHUETTELN UND
 IN EIN COCKTAIL-          GLAS SEIHEN, 3
/6, GIN, 2/6, CURACAO TRIPLE SEC, 1/6, Z
ITRONENSAFT
2130 DATA 0,* LUCAS' LEMON SPECIAL, ZUTA
TEN MIT 2 EL GESTOSSENEM EIS INS    LONG
DRINKGLAS - MIT MINERALWASSER       AUFF
UELLEN, 1 GLAS, ZITRONE, 1 GLAS, GIN, 2
GLAESER, GRAPEFRUIT-SAFT, 1, KIRSCHE
2140 DATA 0,* ADRIA LOOK, ZUTATEN IM MIX
BECHER KURZ SCHUETTELN; INEINE SEKTSCHAL
E GIESSEN UND MIT SEKT AUFFUELLEN, 2 CL,
 GIN, 2 CL, CURACAO, 2 TL, ZITRONENDSAFT
 (NATUR), , SEKT ZUM AUFFUELLEN
2150 DATA 0,* GIN TONIC, ZUTATEN AUF EIS
 GEBEN UND MIT TONIC      AUFGIESSEN, 1 G
LAS, GIN, 1 SCHEIBE, ZITRONE, , TONIC WA
SSER ZUM AUFFUELLEN
2160 DATA 0,* MOON DRIVER, ZUTATEN MIT E
IS MIXEN- IN SEKTSCHALE     GEBEN- MIT SE
KT AUFFUELLEN  GARNIERUNG  1 MARASCHINO-
KIRSCHE + 1 SCHEIBE ZITRONE, 3 CL, GIN,
2 CL, PRINZEN KIRSCH, 1 CL, WERMUT, , SE
KT ZUM AUFFUELLEN
2170 DATA 0,* BOLS FROST, IM TUMBLER EIS
 MIT ZUTATEN VORBEREITEN- MIT TONIC AUFF
UELLEN       GARNIERUNG    SCHEIBE ZITRON
E- ORANGE- KIRSCHE, 4 CL, GIN, 2 CL, CUR
ACAO, 2 CL, ZITRONENSAFT, , TONIC WASSER
 ZUM AUFFUELLEN
2180 DATA 0,* SILVER TOP, ZUTATEN IM MIX
BECHER MIXEN- MIT SEKT IN SEKTSCHALE AUF
FUELLEN   GARNIERUNG       SCHEIBE ORANGE
- KIRSCHE, 1/3, CURACAO TRIPLE SEC, 1/3,
 GIN, 1/6, GRENADINE, 1/6, ORANGENSAFT,
, SEKT ZUM AUFFUELLEN
2190 DATA 0,* BOLS TOP SLING, ZUTATEN IN
 MIXBECHER - 2-3 STUECKE EIS  IN TUMBLER
; UMRUEHREN UND MIT TONIC      AUFFUELLEN
, 4 CL, GIN, 1/2 CL, GRENADINE, 1/2 CL,
CURACAO, A DASH, ANGOSTURA, 2, KIRSCHEN,
 1/2 SCHEIBE, ORANGE
2200 DATA 1/2 SCHEIBE, ZITRONE, , TONIC
WASSER
2210 DATA 0,* HONOLULU COOLER, ZUTATEN M
IXEN  IN AUSGEHOEHLTER ANANAS  SERVIEREN
, 1/3, ANANASSAFT, 1/3, ROSES LIME JUICE
, 1/3, SOUTHERN COMFORT
2220 DATA 0,* PINA COLADA, ZUTATEN MIXEN
 - VERSCHIEDENE RUMSORTEN -UND IN KOKOSN
USS        SERVIEREN, 1/4 L, ANANASSAFT,
```

```
2 EL. KOKOSNUSSCREME. 1 SCHUSS. COINTREA
U. VIEL. RUM
2230 DATA 0.* MAI-TAI. ZUTATEN KRAEFTIG
SCHUETTELN - GARNIERUNGZITRONENSCHEIBE +
     KIRSCHE. 1/2. ZITRONENSAFT. 1/2. RO
SES LIME JUICE. 1 SCHUSS. CURACAO WEISS.
 2 CL. RUM  WEISS. 2 CL. RUM  BRAUN
2240 DATA 0.* PLANTER'S PUNCH. ZUTATEN I
N EINEM HOHEN GLAS MIXEN UND   RVIEREN.
1/3. ORANGENSAFT. 1/3. ZITRONENSAFT. 1/3
. ANANASSAFT. 1 DASH. GRENADINE. 2 CL. R
UM  WEISS. 2 CL. RUM  BRAUN
2250 DATA 0.* MINT-FRAPPE. HOHES GLAS ZU
R HAELFTE MIT GESTOSSENEM  EIS FUELLEN -
 LIKOER DARUEBER LAUFEN     LASSEN - MIT
SODAWASSER AUFFUELLEN       GARNIERUNG FR
ISCHE MINZE. BELIEBIG. PFEFFERMINZLIKOER
2260 DATA 0.* PUSSY-FOOT. EIS + ZUTATEN
MIXEN - MIT FRUECHTEN      GARNIEREN. 1/3
. ANANASSAFT. 1/3. ORANGENSAFT. 1/3. GRA
PEFRUITSAFT
2270 DATA 0.* SUMMERWIND. ZUTATEN + 3 EI
SWUERFEL MIXEN - MIT        BITTER GRAPEFR
UIT AUFFUELLLEN. 1 TL. ZITRONENSAFT. 2 G
LAESER. WODKA. . BITTER GRAPEFRUIT
2280 DATA 0.* BANANA DAIQUIRI. IM ELEKTR
ISCHEN MIXER EIS + EINE HALBE  BANANE MI
T ZUTATEN MIXEN - BIS EIS ZU   SCHNEE WI
RD - IM TUMBLER SERVIEREN. 2 CL. BANANEN
LIKOER. 4 CL. RUM  BRAUN. 1 DASH. ZITRON
E
2290 DATA 0.* CREOLE. IM SHAKER ZUTATEN
+ EIS +                  TRINKSCHOKOLADE MI
XEN  MIT SCHOKOLADEN- STREUSEL SERVIEREN
. 2 CL. CREME DE KAKAO. 4 CL. RUM  BRAUN
. BELIEBIG. TRINKSCHOKOLADE
2300 DATA 0.* BRASIL. ZUTATEN + EIS IN E
ISBECHER GEBEN -      UMRUEHREN - SERVIE
REN. 8 CL. MOCCA   STARK. 1 EL. ZUCKER
BRAUN. 4 CL. RUM
2310 DATA 0.* ZOMBIE. ZUTATEN IN LOGDRIN
KGLAS GEBEN +          FRUECHTE +EIS  VOR
SICHTIG UM-   RUEHREN MIT ZIMTSTANGE SER
VIEREN. 2 CL. KAFFEELIKOER. 4 CL. RUM  B
RAUN
2320 DATA 0.* PRAIRIE OYSTER. NACHEINAND
ER IN EIN COCKTAILGLAS GEBEN. 1 TL. WORC
ESTERSAUCE. 2 TL. KETCHUP. TROPFEN. OEL.
 TROPFEN. ESSIG. 1. EIGELB. . SALZ. . PF
EFFER. . CURRY. . TABASCO. . PAPRIKA. .
MUSKATNUSS
2330 DATA 0.* PORTWEIN COBBLER. HOHES GL
AS ZU 2/3 MIT GESTOSSENEM EIS   FUELLEN
- ZUTATEN - MIT   SODAWASSER AUFFUELLEN
 MIT FRUECHTEN GARNIEREN. 2 TEILE. PORT.
```

```
 1 TEIL. GRAND MARNIER. 1 TEIL. MARASCHI
NO. 1 TL. ZUCKERSIRUP
2340 DATA 0.* SHERRY-COBBLER. COBBLER GL
AS ZU 1/3 MIT FEINEM EIS      FUELLEN -
ZUTATEN - MIT SODA-WASSER  AUFFUELLEN -
DURCHRUEHREN. 2/3. SHERRY  FINO. 1/3. CO
GNAC. 2 SCHUSS. PORT
2350 DATA 0.* RUM CRUSTA. ZUTATEN + 5 ST
UECKE EIS IN SHAKER - 1/2 MINUTE KRAEFTI
G          SCHUETTELN - IN GLAS MIT ZUCKE
RKRUSTE SEIHEN. SAFT 1/2. ZITRONE. 2 EL.
 ZUCKERSIRUP. 2 DASH. ANGOSTURA. 3 DASH.
 MARASCHINO. 1 GLAS. RUM
2360 DATA 0.* BRANDY COOLER. ZUTATEN + E
IS IN SHAKER - MIXEN -        ANSCHLIESSE
ND IN WHISKY-BECHER SEIHEN   UND MIT GIN
GERALE AUFFUELLEN. 1 TL. ZUCKER. SAFT 1/
2. ZITRONE. 1/2 GLAS. BRANDY. . GINGER A
LE
2370 DATA 0.* GIN COOLER. WIE BRANDY COO
LER - JEDOCH GIN STATT BRANDY. .
2380 DATA 0.* EYEBRIGHT COOLER. GROSSER
WHISKYBECHER - ZUTATEN +         3 STUECK
E EIS - MIT GINGER ALE           AUFFUELL
EN - MIT KIRSCHEN UND ANANAS     SERVIERE
N. SAFT 1/2. ZITRONE. 1/2 GLAS. CALVADOS
. 1/2 GLAS. CORDIAL MEDOC. . GINGER ALE
2390 DATA 0.* CABLEGRAM COOLER. GROSSER
WHISKYBECHER - ZUTATEN +         3 STUECK
E EIS - MIT GINGER ALE           AUFFUELL
EN - UMRUEHREN - MIT ZWEI        KIRSCHEN
 SERVIEREN. SAFT 1/2. ZITRONE. 1 EL. GRE
NADINE. 1/2 GLAS. WHISKY. . GINGER ALE
2400 DATA 0.* LOVER'S DREAM. ZUTATEN IM
MIXBECHER MIT GABEL             DURCHSCHLAG
EN - MIT GINGER ALE             AUFFUELLEN
- MIT KIRSCHE UND STROHHALM  SERVIEREN.
1. EI. 1 EL. ZUCKER
2410 DATA SAFT 1/2. ZITRONE. . GINGER AL
E
2420 DATA 0.* CHAMPAGNER DAISY.ZUTATEN I
N ZU 1/3 MIT EIS GEFUELLTEN MIXBECHER GE
BEN- 1/2 MINUTE LANG MIXEN - INSEKTSCHAL
E MIT SEKT AUFFUELLEN - MIT    2 KIRSCHE
N UND 2 STUECKEN ANANAS        SERVIEREN
.SAFT 1/4.ZITRONE.1 TL.GRENADINE
2430 DATA 1/2 GLAS.BRANDY..SEKT
2440 DATA 0.* BRANDY DAISY. WIE CHAMPAGN
ER DAISY - JEDOCH MIT 1      STATT 1/2 GL
AS BRANDY - MIT SODA         AUFFUELLEN.
.
2450 DATA 0.* GIN DAISY. WIE CHAMPAGNER
DAISY - JEDOCH GIN STATT BRANDY. SAFT 1/
4. ZITRONE. 1 TL. GRENADINE. 1/2 GLAS. G
IN
```

```
2460 DATA 0,* RUM DAISY, WIE CHAMPAGNER
DAISY - JEDOCH RUM STATT BRANDY, SAFT 1/
4, ZITRONE, 1 TL, GRENADINE, 1/2 GLAS, R
UM
2470 DATA 0,* WHISKY DAISY, WIE CHAMPAGN
ER DAISY - JEDOCH WHISKY   STATT BRANDY
, SAFT 1/4, ZITRONE, 1 TL, GRENADINE, 1/
2 GLAS, WHISKY
2480 DATA 0,* ASCOT DAISY, MIXBECHER ZU
1/3 MIT EIS FUELLEN -        ZUTATEN - SCH
UETTELN UND IN SEKTSCHALE  GEBEN - MIT 5
 KIRSCHEN  GARNIEREN - SODAHINZUFUEGEN,
2 TL, CHERRY BRANDY, SAFT 1/4, ZITRONE,
2 TL, WHISKY, 2 EL, GIN
2490 DATA 0,* MANHATTAN, MIXEN - MIT KIR
SCHE GARNIEREN, 1 DASH, ANGOSTURA, 2/3,
WHISKY, 1/3, WERMUT  ROT
2500 DATA 0,* BRONX, MIXEN, 1/4, GIN, 1/
4, WERMUT  ROT, 1/4, WERMUT  WEISS, 1/4,
 ORANGENSAFT
2510 DATA 0,* INDIA, MIXEN, 1 DASH, MARA
SCHINO, 1/3, GRAPEFRUITSAFT, 2/3, GIN
2520 DATA 0,* MADAME, MIXEN - MIT ZITRON
ENSPIRALE UND KIRSCHE GARNIEREN, 1 DASH,
 ANGOSTURA, 2 SCHUSS, WERMUT, 4 CL, GIN
2530 DATA 0,* MOONLIGHT, MIXEN, 1 DASH,
ORANGENSAFT, 1 DASH, CURACAO, 1/3, WERMU
T, 2/3, GIN
2540 DATA 0,* PARADIES, MIXEN, 1/3, GIN,
 1/3, APRICOT BRANDY, 1/3, ORANGENSAFT
2550 DATA 0,* EXPRESS, MIXEN, 1/3, GIN,
1/3, APRICOT BRANDY, 1/3, WERMUT
2560 DATA 0,* BAMBOO, MIXEN - MIT 1 STUE
CK ZITRONENSCHALE     SERVIEREN, 2 DASH,
 ANGOSTURA, 1/2, SHERRY, 1/2, WERMUT
2570 DATA 0,* KIR ROYAL, ZUCKER IN SEKTG
LAS - ZUTATEN DER REIHE  NACH BEIGEBEN,
1 TL, ZUCKER, 1 DASH, ANGOSTURA, 2 CL, C
OGNAC, , SEKT ZUM AUFFUELLEN
2580 DATA 0,* OHIO, MIT KALTEM SEKT AUFF
UELLEN - KIRSCHE EINLEGEN, 1 DASH, ANGOS
TURA, 1/3, WHISKY, 1/3, WERMUT  ROT, 1/3
, CURACAO
2590 DATA 0,* TOM COLLINS, 4 EISWUERFEL
IN LONGDRINKGLAS - ZUTATEN - MIXEN - MIT
 SODA       AUFFUELLEN, SAFT 1, ZITRONE,
2 TL, ZUCKER, 2 CL, GIN
2600 DATA 0,* WHISKY FIX, GLAS ZU 1/3 MI
T EIS FUELLEN - ZUTATEN - UMRUEHREN - ZI
TRONEN-    SCHEIBE UND      STROHHALM BEIG
EBEN, 2 EL, ZUCKERSIRUP, SAFT 1, ZITRONE
, 1 CL, CURACAO ORANGE, 2 CL, WHISKY  SC
OTCH
2610 DATA 0,* PRINCE OF WALES, HOHES GLA
S - 4 EISWUERFEL - ZUTATEN -   UMRUEHREN
 UND MIT SEKT     AUFFUELLEN -  GARNIERUN
G  1 SCHEIBE ORANGE, 1/2 GLAS, CURACAO,
2 DASH, ANGOSTURA
```

```
2620 DATA O.* SALTY DOG. GLAS ZUR HAELFT
E MIT EIS FUELLEN -      ZUTATEN + 1 PRI
SE SALZ -     UMRUEHREN -MIT STROHHALM S
ERVIEREN, 2 CL, WODKA, 4 CL, GRAPEFRUITS
AFT
2630 DATA O.* PINK CARTER PICK ME UP. SE
KTSCHALE - 2 EISWUERFEL + ZUTATEN - MIT
ZITRONENSCHALE         ABSPRITZEN. 3 DASH
. ANGOSTURA. 1/4 GLAS. GRENADINE. 1/4 GL
AS. ZITRONENSAFT. 1/2 GLAS. GIN. , SEKT
ZUM AUFFUELLEN
2640 DATA O.* WODKA FIZZ. ZUTATEN MIT EI
S GUT SCHUETTELN - IN GLASSEIHEN - MIT S
ODA AUF- FUELLEN. 1/4. WODKA. 3/4. ANANA
SSAFT
2650 DATA O.* BLOODY MARY. ZUTATEN MIT E
IS SCHUETTELN. , TOMATENSAFTGUT GEKUEHLT
, 2 CL. WODKA, 1 DASH. WORCESTER-SAUCE.
, PFEFFER, . SALZ. , PAPRIKA. 1 DASH. ZI
TRONENSAFT
2660 DATA O.* SCREW DRIVER. WODKA IN EIN
 GLAS MIT EIS GEBEN - MIT   SAFT AUFFUEL
LEN -          SCHUETTELN, 2 CL. WODKA, .
 ORANGENSAFT ZUM AUFFUELLEN
2670 DATA O.* MOSCOW MULE. ZUTATEN + EIS
 INS GLAS - MIT GINGER AUF-FUELLEN. 4 CL
. WODKA, . GINGER ALE ZUM AUFFUELLEN, 1.
 ZITRONENSCHEIBE
2680 DATA O.* GIMLET. ZUTATEN IM SHAKER
MIT EIS SCHUETTELN. 1/3. ROSES LIME JUIC
E. 2/3. WODKA
2690 DATA O.*ANGOSTURA SCORPION. MIT EIS
 MIXEN. 2 EL. ZUCKERSIRUP. 2 EL. ROSES L
IME JUICE. 1 DASH. ANGOSTURA. 3 EL. RUM
2700 DATA O.* RUM PUNCH. ZUSAMMENGEBEN -
 BITTER & GINGER ALE ALS LETZTES. 1 TASS
E, ZITRONENSAFT, 2 TASSEN. ERDBEERSIRUP.
 3 TASSEN. RUM WEISS   (JAMAICA). 2 TASS
EN. ANANASSAFT. 1 TASSE. GRAPEFRUITSAFT.
 1 TASSE. GINGER ALE. DASHES. ANGOSTURA
2710 DATA O.* INCIDENT. SORBET + ZUTATEN
 IN SCHMALES GLAS GEBEN. 1 EL. ZITRONENS
ORBET. 4 CL. RUM  BRAUN  EISKALT. 4 CL.
ROTWEIN
2720 DATA O.* WHITE WITCH. ZUTATEN SCHAU
MIG MIXEN. 1/4. KAKAO RUM LIKOER. 1/4. R
UM. 1/2. VANILLE EIS
2730 DATA O.* MORGAN'S REVENGE. MIT EIS
MIXEN - MIT SODA AUFFUELLEN. 1/4. RUM. 1
/2. ANANASSAFT, 1/4. ORANGENLIKOER  BITT
ER
2740 DATA O.* PEACH'S SPECIAL. EIS + ZUT
ATEN IM SHAKER MIXEN - MIT    KIRSCHE S
ERVIEREN. 1 DASH. ANGOSTURA. 1/3. CHERRY
```

```
 BRANDY, 1/3, WERMUT  WEISS, 1/3, GIN
2750 DATA 0,* ACAPULCO, 4 ZERKLEINERTE E
ISWUERFEL IN GLAS - MIT SODA AUFFUELLEN,
 1 TL, ZUCKERSIRUP, 5 CL, TEQUILA, 2.5 C
L, CASSIS, 1, ZITRONENSCHEIBE, , SODA ZU
M AUFFUELLEN
2760 DATA 0,* TEQUILA CALIENTE, ZUTATEN
MIXEN & 3 EISWUERFEL MIXEN - MITSODA AUF
FUELLEN, 4 CL, TEQUILA, 1 CL, CASSIS, 1
CL, ROSES LIME JUICE, 2 DASHES, GRENADIN
E, 5-10 CL, SODA
2770 DATA 0,* THE BLACKEST RUSSIAN, ZUTA
TEN IN GLAS GEBEN - VORSICHTIG 3 EISWUER
FEL ZUGEBEN, 2.5 CL, KAFFEELIKOER, 7.5 C
L, WODKA
2780 DATA 0,* WERMUT CASSIS, ZUTATEN & 3
 EISWUERFEL IN GLAS GEBEN -  MIT SODA AU
FFUELLEN, 8 CL, WERMUT  TROCKEN, 3.5 CL,
 CASSIS
2790 DATA 0,* WHISKY COCKTAIL, ZUTATEN &
 3 EISWUERFEL IM SHAKER MIXEN, 5 CL, WHI
SKY, 2 DASHES, ANGUSTURA, 1 TL, ZUCKERSI
RUP, 1, KIRSCHE
2800 DATA 0,* ZUBROWKATINI, ZUTATEN & 3
EISWUERFEL IM SHAKER MIXEN, 4 CL, WODKA,
 6 CL, PFEFFERMINZLIKOER  WEISS, 4 CL, Z
ITRONENSAFT, 1, KIRSCHE  GRUEN
2810 DATA 0,* ADONIS, ZUTATEN & 3 EISWUE
RFEL VERRUEHREN, 1.5 CL, WERMUT  ROT, 3
CL, SHERRY, 1 DASH, ANGOSTURA
2820 DATA 0,* HONEYMOON, 3 EISWUERFEL &
ZUTATEN IM SHAKER MIXEN, 3 DASHES, CURAC
AO ORANGE, 1 CL, ORANGENSAFT, 1.5 CL, BE
NEDICTINE, 2.5 CL, CALVADOS
2830 DATA 0,* RED SHADOW, 3 EISWUERFEL &
 ZUTATEN IM SHAKER MIXEN, 3 DASHES, ANGO
STURA, 1 CL, APRICOT BRANDY, 1 CL, CHERR
Y BRANDY, 3 CL, WHISKY
2840 DATA 0,* TAKE TWO, 3 EISWUERFEL & Z
UTATEN MIXEN, 2.5 CL, GIN, 1.5 CL, COINT
REAU, 1 CL, CAMPARI
2850 DATA 0,* ROLLS ROYCE, ZUTATEN IN GE
GEBENER REIHENFOLGE IN EIN GLAS GEBEN -
UMRUEHREN, 2-3, EISWUERFEL, 2.5 CL, GIN,
 1.5 CL, WERMUT  DRY, 1 CL, WERMUT  WEIS
S, 1-2 DASHES, BENEDICTINE, 1, KIRSCHE
2860 FOR I=1 TO 300
2870 NEXT I
2880 RESUME 200
```

Ein Maschinensprache-Monitor in BASIC

Dieses Programm stammt aus dem 'Colour-Genie-Buch 1'. Es wird in diesem Buch nochmals gelistet, da der Monitor um Routinen zum Disketten-Sektor lesen und schreiben erweitert wurde. Für genaue Erklärungen des Monitors schlagen Sie bitte im 'Colour-Genie-Buch 1' nach, die Erklärung der Diskettenroutinen finden ein paar Seiten weiter.
Nun noch eine Erklärung der einzelnen Monitor-Kommandos:

1. Hex-Dump
   Eingabe: Hadresse
   Ab der eingegebenen Adresse wird der Speicherinhalt in hexadezimal gelistet. Jeweils nach Ausgabe einer Zeile wird die Tastatur auf folgende Tasten abgefragt:
   <CLEAR> - Listing abbrechen.
   <Leertaste> - Listing anhalten, bis <RETURN> gedrückt wird.
   Die einzelnen Zeilen haben folgendes Format:
   Die ersten 4 Zeichen geben die Adresse an, in der das erste der 8 Bytes, die in dieser Zeile gelistet werden, steht. Dann folgen die 8 Bytes und zum Schluß, etwas abgesetzt, folgt die sogenannte Checksum. Dieses Byte ist das nieder--wertige Byte der Summe aller 8 Bytes und dient als Prüfmöglichkeit für die Richtigkeit einer Eingabe. Alle Maschinenprogramme in diesem Buch werden als Hex-Dump mit Prüfsumme gelistet, so daß Sie eine einfache Möglichkeit haben, die Richtigkeit Ihrer Eingabe zu prüfen.
2. ASCII-Dump
   Eingabe: Aadresse
   Ab der eingegebenen Adresse wird der Speicherinhalt als ASCII-Zeichen gelistet. Jede Zeile beginnt mit der Adresse, in der das erste der 24 folgenden Zeichen steht, dann folgen die ASCII-Zeichen. Jedes Byte unter 32 dez. wird als '.' dargestellt, um die Bildschirmausgabe nicht durch Codes wie 'Backspace' etc. zu zerstören. Auch beim ASCII-Dump kann die Ausgabe durch <CLEAR> abgebrochen und durch <Leertaste> angehalten werden.
3. Edit hexadezimal
   Eingabe: Eadresse
   Ab der eingegebenen Adresse kann der Speicherinhalt verändert werden. Es wird jeweils die Adresse und der bisherige Inhalt dieser Speicherstelle ausgegeben. Sie können dann ein neues Byte eingeben, das dann in die Speicherstelle abgelegt wird. Wenn Sie statt einer Eingabe nur <RETURN> drücken, wird der Editmodus abgebrochen.

4. Edit ASCII
   Eingabe: Tadresse
   Ab der eingegebenen Adresse können Sie Texte direkt in den Speicher eingeben. Die Adresse wird angegeben, ihr Inhalt wird in hexadezimal und ASCII (Code kleiner als 32 = '.') ausgegeben. Sie können nun ein Zeichen eingeben, das an dieser Speicherstelle gespeichert wird. Auch hier wird die Eingabe abgebrochen, wenn Sie nur <RETURN> drücken. Achtung: Um ein Leerzeichen in den Speicher einzugeben, müssen Sie folgendes eingeben: " ". Dies liegt daran, daß der INPUT-Befehl des Colour-Basic alle führenden Leerzeichen ignoriert, eine Eingabe eines Leerzeichens allein also als Leerstring angesehen wird.
5. Go to memory
   Eingabe: Gadresse byte
   Es wird zu einem Programm gesprungen, das bei der eingegebenen Adresse beginnt. Wenn Ihr Programm mit einem CALL 0A7FH beginnt, so können Sie ein Byte in das L-Register übergeben. Wenn Sie Ihr Programm mit einem JP 0A9AH beenden, so wird der Inhalt des L-Registers wieder an das Basic übergeben und der Monitor gibt es hexadezimal aus.
6. Hexadezimal Rechnung
   Eingabe: Azahl1 zahl2
   Zahl1, Zahl2, ihre Summe und ihre Differenz werden in hexadezimal und dezimal ausgegeben.
7. SYSTEM-Band schreiben
   Eingabe: Wstart ende einsprung
   Zusätzlich wird noch der Programmname abgefragt. Dieser Name darf maximal 6 Zeichen lang sein und das erste Zeichen muß ein Buchstabe sein. Dann wird der Bereich von start bis ende im SYSTEM-Band-Format auf Kassette geschrieben. Um dieses Band wieder einzulesen, benutzen Sie den SYSTEM-Befehl des Colour-Basic und geben Sie den von Ihnen gewählten Programmnamen an.
8. Laufwerk selektieren
   Eingabe: Slaufwerksnummer
   Das angegebene Laufwerk (0 <= Laufwerksnummer <= 4) wird angesprochen. Alle Diskettenzugriffe sind nun auf dieses Laufwerk gerichtet.
9. Sektor lesen/schreiben
   Eingabe: Dadresse disk-relativer-sektor flag
   flag=1: Der angegebene Sektor wird ab der angegebenen Adresse in den Speicher gelesen.
   flag=2: Die ab der angegebenen Adresse im Speicher stehenden 256 Bytes werden in den angegebenen Sektor geschrieben.

flag=3: Wie flag=2, der beschriebene Sektor wird aber als Inhaltsverzeichnis-Sektor gekennzeichnet.

Anmerkung: Wenn Sie einen Sektor des Inhaltsverzeichnisses einlesen, wird 'Disk error No. 6' ausgegeben. Dies ist kein (!) Fehler, sondern Sie können so diese Sektoren erkennen. Wenn Sie einen solchen Sektor wieder auf die Diskette schreiben, müssen Sie flag=3 angeben, sonst ist das Inhaltverzeichnis nicht mehr lesbar!!!!

Wenn Sie statt Hadresse bzw. Aadresse PHadresse bzw. PAadresse angeben, geht das Listing parallel auf den Drucker.

Programmlisting:

```
10 CLS:CLEAR1000
20 PRINT"Command";:INPUTA$
30 IFA$=""THEN20
40 GOSUB670
50 IFB$="P"THENPR=1:GOSUB670ELSEPR=0
60 C$="HAETGASDW"
70 FORA=1TOLEN(C$):IFB$<>MID$(C$,A,1)THENNEXTA:GOTO20
80 ONAGOTO100,200,310,360,430,500,790,810,940
90 GOTO20
100 GOSUB690
110 X=Z
120 Z=X:GOSUB750:PRINTZ$;" ";:IFPR=1THENLPRINTZ$;" ";
130 C=0:FORY=0TO7:Z=PEEK(X+Y):C=C+Z:GOSUB750:PRINTRIGHT$(Z$,2);" ";:IF
PR=1THENLPRINTRIGHT$(Z$,2);" ";
140 NEXTY:PRINT"  ";:IFPR=1THENLPRINT"  ";
150 Z=C:GOSUB750:PRINTRIGHT$(Z$,2):IFPR=1THENLPRINTRIGHT$(Z$,2)
160 X=X+8
170 KB=PEEK(&HF840):IFKB=2THEN20
180 IFKB<>128THEN120
190 KB=PEEK(&HF840):IFKB<>1THEN190ELSE120
200 GOSUB690
210 X=Z
220 Z=X:GOSUB750:PRINTZ$;" ";:IFPR=1THENLPRINTZ$;" ";
230 FORY=0TO23:Z=PEEK(X+Y)
240 IFZ<32THENZ=ASC(".")
250 PRINTCHR$(Z);:IFPR=1THENLPRINTCHR$(Z);
260 NEXTY:PRINT:IFPR=1THENLPRINT
270 X=X+24
280 KB=PEEK(&HF840):IFKB=2THEN20
290 IFKB<>128THEN220
300 KB=PEEK(&HF840):IFKB<>1THEN300ELSE220
310 GOSUB690
320 X=Z
330 Z=X:GOSUB750:PRINTZ$;" - ";:Z=PEEK(X):GOSUB750:PRINTRIGHT$(Z$,2);"
 - ";
340 A$="":INPUTA$:IFA$=""THEN20
350 GOSUB690:POKEX,ZAND255:X=X+1:GOTO330
360 GOSUB690
370 X=Z
380 Z=X:GOSUB750:PRINTZ$;" - ";:Z=PEEK(X):GOSUB750:PRINTRIGHT$(Z$,2);"
 (";
390 IFZ<32THENZ=ASC(".")
400 PRINTCHR$(Z);") - ";
```

```
410 A$="":INPUTA$:IFA$=""THEN20
420 POKEX,ASC(A$):X=X+1:GOTO380
430 GOSUB690
440 Z1=INT(Z/256)AND255:Z2=ZAND255
450 POKE16526,Z2:POKE16527,Z1
460 GOSUB670:IFB$=" "THEN460ELSEA$=B$+A$
470 GOSUB690:Z=ZAND255
480 Z=USR(Z)
490 GOSUB750:PRINTRIGHT$(Z$,2):GOTO20
500 GOSUB690:IFZ>=0THENX1=ZELSEX1=Z+65536
510 GOSUB670:IFB$=" "THEN510ELSEA$=B$+A$
520 GOSUB690:IFZ>=0THENX2=ZELSEX2=Z+65536
530 PRINT"  A     B    A+B   A-B"
540 PRINT" ";:Z=X1:GOSUB750:PRINTZ$;"  ";:Z=X2:GOSUB750:PRINTZ$;"  ";
550 Z=X1+X2:IFZ>65535THENZ=Z-65536
560 GOSUB750:PRINTZ$;"  ";
570 Z=X1-X2:IFZ<0THENZ=Z+65536
580 GOSUB750:PRINTZ$
590 Z=X1:GOSUB640:PRINTZ$;" ";:Z=X2:GOSUB640:PRINTZ$;" ";
600 Z=X1+X2:IFZ>65535THENZ=Z-65536
610 GOSUB640:PRINTZ$;" ";
620 Z=X1-X2:IFZ<0THENZ=Z+65536
630 GOSUB640:PRINTZ$:GOTO20
640 Z$=STR$(Z):Z$=RIGHT$(Z$,LEN(Z$)-1)
650 IFLEN(Z$)<5THENZ$=STRING$(5-LEN(Z$),32)+Z$
660 RETURN
670 B$="":IFLEN(A$)=0THENRETURN
680 B$=LEFT$(A$,1):A$=RIGHT$(A$,LEN(A$)-1):RETURN
690 Z=0
700 GOSUB670:IFB$<"0"ORB$>"F"THEN740
710 IFB$>"9"ANDB$<"A"THEN740
720 IFB$<="9"THENZ1=ASC(B$)-48ELSEZ1=ASC(B$)-55
730 Z=Z*16+Z1:GOTO700
740 IFZ>32767THENZ=Z-65536
750 Z1=INT(Z/256)AND255:Z2=ZAND255:Z$=""
760 Z3=INT(Z1/16):GOSUB770:Z3=Z1AND15:GOSUB770:Z3=INT(Z2/16):GOSUB770:
Z3=Z2AND15
770 IFZ3<10THENZ$=Z$+CHR$(Z3+48)ELSEZ$=Z$+CHR$(Z3+55)
780 RETURN
790 GOSUB670:IFASC(B$)<48ORASC(B$)>51THEN20
800 POKE&H5A08,ASC(B$)-48:CALLD0C7:GOTO20
810 GOSUB690:AD=Z:GOSUB690:SN=Z:GOSUB690:FL=Z
820 RESTORE
830 READA$:IFA$<>"SECIO"THEN830
840 A$="":FORX=1TO15:READA:A$=A$+CHR$(A):NEXTX:FORX=1TOFL:READDR:NEXTX
850 X=VARPTR(A$):X=PEEK(X+1)+256*PEEK(X+2):IFX>32767THENX=X-65536
860 POKE X+1,(AD AND 255):POKE X+2,(INT(AD/256)AND255)
870 POKE X+4,(SN AND 255):POKE X+5,(INT(SN/256)AND255)
880 POKE X+7,(DR AND 255):POKE X+8,(INT(DR/256)AND255)
890 DEFUSR1=X:X=USR1(0):IFX=0THEN20
900 PRINT"Disk Error No.";X:GOTO20
910 DATA SECIO
920 DATA 33,0,0,17,0,0,205,0,0,111,38,0,195,154,10
930 DATA-12433,-12417,-12421
940 RESTORE
950 WR$=""
960 READXX$:IF XX$<>"SYSTEM" THEN 960
970 FORX=1TO127
980 READA:WR$=WR$+CHR$(A)
990 NEXTX:X=VARPTR(WR$):X=PEEK(X+1)+256*PEEK(X+2):IFX>32767THENX=X-655
36
1000 R=X+95:R1=INT(R/256)AND255:R2=RAND255
```

```
1010 POKEX+66,R2:POKEX+67,R1:POKEX+75,R2:POKEX+76,R1
1020 GOSUB1220:GOSUB1260:GOSUB1220:GOSUB1260:GOSUB1220
1030 PRINT"Filename:";:INPUTPN$
1040 FORX1=XTOX+5:POKEX1,32:NEXTX1
1050 FORX1=1TOLEN(PN$):POKEX-1+X1,ASC(MID$(PN$,X1,1)):NEXTX1
1060 X=X+6:X1=INT(X/256)AND255:X2=XAND255
1070 POKE16526,X2:POKE16527,X1
1080 ONERRORGOTO1100
1090 DEFUSR0=X
1100 ONERRORGOTO0
1110 ONERRORGOTO1130
1120 CMD"T
1130 ONERRORGOTO0
1140 X1=USR(0)
1150 GOTO20
1160 DATA SYSTEM
1170 DATA0,0,0,0,0,0,32,32,32,32,32,32
1180 DATA42,142,64,17,250,255,25,205,63,2,62,85,205,31,2,6,6,126,35,20
5,31,2,16,249,17,244,255,25,229,221,225,221,94,0,221,86,1,221,110,2,22
1,102,3
1190 DATA175,237,82,35,124,183,40,8,6,0,205,0,0,37,24,244,125,69,183,1
96,0,0,62,120,205,31,2
1200 DATA221,126,4,205,31,2,221,126,5,205,31,2,201
1210 DATA62,60,205,31,2,120,205,31,2,123,205,31,2,79,122,205,31,2,129,
79,26,19,205,31,2,129,79,16,247,195,31,2
1220 GOSUB690
1230 Z1=INT(Z/256)AND255
1240 Z2=ZAND255
1250 POKEX,Z2:POKEX+1,Z1:X=X+2:RETURN
1260 GOSUB670
1270 IFB$=" "THEN1260
1280 A$=B$+A$:RETURN
```

Ein Grafikprogramm in Maschinensprache: Lissajou-Figuren

Dieses Programm demonstriert die Möglichkeiten mit Maschinensprache schnelle Grafiken anzulegen.
Es berechnet und zeichnet Lissajou-Figuren in den Grafikspeicher.
Da die Figuren aus zwei phasenverschobenen Sinuskurven entstehen, wurde im Programm (ab &H7200) eine Tabelle mit Sinuswerten, die auf die Grafikauflösung abgestimmt sind angelegt. Damit werden lange Berechnungen vermieden.
Aus der Tabelle werden einfach für jeden Punkt zwei Koordinaten entnommen, die einen einstellbaren Abstand voneinander haben.

Geben Sie das Programm mit dem weiter vorne gelisteten Basic-Monitor ein. Wenn Ihr Colour-Genie nur mit 16 K-Byte ausgerüstet ist, müssen Sie vorher bei Mem Size ? mit 28672 antworten.
Nachdem Sie es eingegeben haben, sollten Sie das Programm zunächst auf Kassette oder Diskette abspeichern.
Die Anfangsadresse ist &H7000, Endadresse &H7437 und Startadresse ist &H7090.

Falls Sie einen Monitor mit Disassembler zur Verfügung haben, können Sie die Einzehlheiten des Programmes leicht analysieren, da es modular aufgebaut ist.

Noch ein Wort zur Bedienung:
Mit den Pfeiltasten können Sie das Frequenzverhältnis (Pfeil-oben & Pfeil-unten) bzw. die Geschwindigkeit der Phasenverschiebung einstellen (Pfeil-links & Pfeil-rechts).

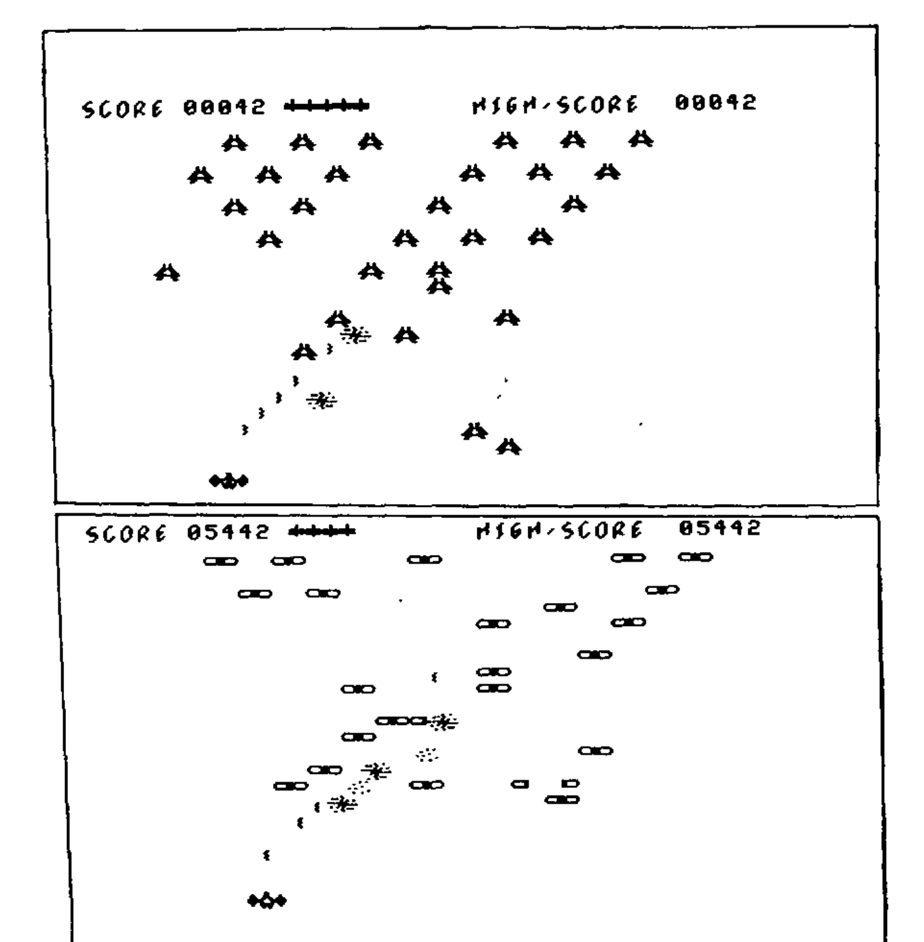
Auf den nächsten drei Seiten finden Sie das Hexlisting des Programms mit Prüfsummen.

```
7000 E5 D5 21 00 72 5F 16 00    C2
7008 19 7E D1 E1 C9 00 00 00    12
7010 4F 3A FF 7F 81 E6 7F C3    B0
7018 00 70 00 00 00 00 00 00    70
7020 06 7F C5 78 CD 00 70 C6    C5
7028 20 67 3A FE 7F 4F 3A FD    C4
7030 7F 81 32 FE 7F CD 10 70    FC
7038 6F CD C6 3C C1 05 C2 22    E8
7040 70 C9 00 00 00 00 00 00    39
7048 00 00 00 00 00 00 00 00    00
7050 3A FF 7F 4F 3A FC 7F 81    3D
7058 32 FF 7F CD 20 70 21 FD    2B
7060 7F 3A 40 F8 FE 08 20 01    18
7068 34 FE 10 20 01 35 2B FE    C1
7070 40 20 01 34 FE 20 20 01    D4
7078 35 CD 40 71 C3 D0 70 00    B6
7080 00 00 00 00 00 00 00 00    00
7088 00 00 00 00 00 00 00 00    00
7090 3E 03 32 13 43 3E 02 32    3B
7098 FC 7F 32 FD 7F 21 00 48    92
70A0 11 01 48 01 FF 0F 36 00    9F
70A8 ED B0 C3 EB 70 00 00 00    8B
70B0 E5 D5 06 05 7E 12 23 C5    3D
70B8 01 28 00 EB 09 EB C1 10    D9
70C0 F3 D1 E1 C9 00 00 00 00    6E
70C8 00 00 00 00 00 00 00 00    00
70D0 21 00 48 11 00 58 01 48    1B
70D8 0D ED B0 21 00 48 11 01    25
70E0 48 01 48 0D 36 00 ED B0    71
70E8 C3 50 70 3E 18 32 03 43    51
70F0 3E 60 32 09 43 CD A9 38    CA
70F8 C3 6B 71 00 00 00 00 00    9F
7100 00 A8 88 88 88 A8 20 20    28
7108 20 20 20 A8 08 A8 80 A8    E0
7110 A8 08 28 08 A8 88 88 A8    40
7118 08 08 A8 80 A8 08 A8 A8    38
7120 80 A8 88 A8 A8 08 08 08    18
7128 08 A8 88 A8 88 A8 A8 88    40
7130 A8 08 A8 00 00 00 00 00    58
7138 00 00 00 00 00 00 00 00    00
7140 3A FC 7F 11 28 48 0E 64    A8
7148 CD 80 71 0E 0A CD 80 71    94
7150 0E 01 CD 80 71 3A FD 7F    83
7158 11 40 49 0E 64 CD 80 71    CA
7160 0E 0A CD 80 71 0E 01 CD    B2
7168 80 71 C9 21 80 72 11 48    26
7170 65 01 FF 01 ED B0 C3 50    16
7178 70 00 00 00 00 00 00 00    70
7180 D5 21 01 71 11 05 00 19    97
7188 91 30 FC 81 B7 ED 52 D1    05
7190 13 F5 CD 80 70 F1 C9 00    AF
7198 00 00 00 00 00 00 00 00    00
71A0 00 00 00 00 00 00 00 00    00
```

71A8	00	00	00	00	00	00	00	00	00
71B0	00	00	00	00	00	00	00	00	00
71B8	00	00	00	00	00	00	00	00	00
71C0	00	00	00	00	00	00	00	00	00
71C8	00	00	00	00	00	00	00	00	00
71D0	00	00	00	00	00	00	00	00	00
71D8	00	00	00	00	00	00	00	00	00
71E0	00	00	00	00	00	00	00	00	00
71E8	00	00	00	00	00	00	00	00	00
71F0	00	00	00	00	00	00	00	00	00
71F8	00	00	00	00	00	00	00	00	00
7200	30	31	33	34	36	37	38	3A	A7
7208	3B	3D	3E	3F	41	42	43	44	FF
7210	45	46	47	48	49	4A	4B	4C	44
7218	4C	4D	4D	4E	4E	4E	4E	4E	6C
7220	4F	4E	4E	4E	4E	4E	4D	4D	6F
7228	4C	4C	4B	4A	49	48	47	46	4B
7230	45	44	43	42	41	3F	3E	3D	09
7238	3B	3A	38	37	36	34	33	31	B2
7240	30	2E	2C	2B	29	28	27	25	52
7248	24	22	21	20	1E	1D	1C	1B	F9
7250	1A	19	18	17	16	15	14	13	B4
7258	13	12	12	11	11	11	11	11	8C
7260	11	11	11	11	11	11	12	12	8A
7268	13	13	14	15	16	17	18	19	AD
7270	1A	1B	1C	1D	1E	20	21	22	EF
7278	24	25	27	28	29	2B	2C	2E	46
7280	00	00	00	00	00	14	00	01	15
7288	55	40	05	55	00	05	55	00	49
7290	01	55	00	15	55	50	05	55	6A
7298	00	50	05	00	01	54	00	40	EA
72A0	15	00	15	00	15	00	00	00	3F
72A8	00	00	00	00	00	14	00	01	15
72B0	55	40	15	55	40	15	55	40	E9
72B8	05	55	40	15	55	50	15	55	BE
72C0	40	50	05	00	04	01	01	40	DB
72C8	51	40	51	40	51	40	00	00	B3
72D0	01	00	00	01	00	14	00	00	16
72D8	14	00	54	01	40	54	01	40	3E
72E0	15	01	50	14	01	50	15	05	E5
72E8	40	50	05	00	04	51	05	40	2F
72F0	51	40	51	40	01	40	00	00	63
72F8	04	40	00	04	40	14	00	00	9C
7300	14	00	54	00	00	54	00	00	BC
7308	14	00	50	00	00	50	14	01	C9
7310	40	50	05	00	04	41	01	40	1B
7318	15	40	15	00	05	00	00	00	6F
7320	04	10	00	10	40	14	00	00	78
7328	14	00	15	55	00	15	55	00	E8
7330	15	55	50	00	00	50	14	01	1F
7338	40	50	05	00	04	51	01	40	2B
7340	01	40	51	40	01	40	00	00	13

7348	04	04	00	40	40	14	00	00	9C
7350	14	00	05	55	40	05	55	40	48
7358	15	55	50	00	00	50	14	01	1F
7360	40	50	05	00	04	01	01	40	DB
7368	51	40	51	40	51	40	00	00	B3
7370	01	01	01	01	00	14	00	00	18
7378	14	00	00	05	50	00	01	50	BA
7380	14	00	50	00	00	50	14	01	C9
7388	40	50	05	00	01	54	05	50	3F
7390	15	00	15	00	15	00	00	00	3F
7398	01	00	44	01	00	14	00	00	5A
73A0	14	00	00	01	50	00	00	50	B5
73A8	14	00	50	00	00	50	14	01	C9
73B0	40	50	05	00	04	00	00	00	99
73B8	00	00	00	00	00	00	00	00	00
73C0	00	40	10	04	00	14	00	00	68
73C8	14	00	14	05	50	14	01	50	E2
73D0	14	00	50	14	01	50	15	05	E3
73D8	40	54	15	00	05	55	55	55	AD
73E0	55	55	55	55	55	40	00	00	E9
73E8	00	40	44	04	00	15	55	41	33
73F0	55	40	15	55	40	15	55	40	E9
73F8	14	00	50	15	55	50	15	55	88
7400	40	55	55	00	00	00	00	00	EA
7408	00	00	00	00	00	00	00	00	00
7410	00	15	01	50	00	15	55	41	11
7418	55	40	05	55	00	05	55	00	49
7420	14	00	50	05	55	40	05	55	58
7428	00	15	54	00	05	55	55	55	6D
7430	55	55	55	55	55	40	00	00	E9

ZALAGA



EIN SUPER-MASCHINENSPRACHE-SPIEL
KOMPLETT GELISTET UND ERKLAERT !

'ZALAGA' ein Spielprogramm in Maschinensprache
fuer das Colour-Genie mit 16KByte RAM.

ZALAGA ist ein Action-Spiel in Maschinensprache, das auf Geschwindigkeit ausgelegt ist.
Es soll einmal demonstrieren, wo die wahren Vorteile der Maschinenprogrammierung liegen.
Sie geben das Programm am guenstigsten mit dem weiter vorne gelisteten Basic-Monitor ein oder mit einem anderen Maschinensprachemonitor. Wenn Ihr Geraet nur mit 16KByte RAM ausgeruestet ist, so geben Sie vorher bei MEM SIZE? 28672 ein und laden dann den Monitor.
Das Programm besteht aus zwei grossen Bloecken:
Das eigentlicAe Programm belegt den Speicher von 7000H bis 75FFH. Es ist wiederum in kleinere Unterprogramme eingeteilt, wie aus der Dokumentation hervorgeht.
Der zweite Teil sind Tabellen, Anfangsgrafik und die Werte fuer die definierbaren Zeichen.
Dieser Block beginnt bei 7800H. Sie haben also noch genuegend freien Speicher um das Programm um- oder auszubauen.
Die Luecken zwischen den einzelnen Unterprogrammen erlauben kleine Aenderungen auch innerhalb des ersten Programmteils. Auf jedenfall sollten Sie zuerst eine Original-Version auf Kassette oder Diskette sichern.
Also speichern Sie von 7000H bis 7FFFH mit der Start-adresse 75B0H.
Noch einige Anmerkungen ueber den Aufbau der Tabellen.
Die Tabelle der Positionen der Feinde besteht aus 128 Bytes. Fuer jedes der 32 Objekte stehen 4 Bytes zur Verfuegung. Die ersten beiden Bytes entsprechen der Position des Feindes im Bildschirmspeicher (4400H-47E7H).
Ist diese Position gleich 0, so bedeutet dies, dass der jeweilige Feind schon abgeschossen wurde.
Das 3. Byte beinhaltet das LSB der Y-Richtung des Feindes.
Es kann die Werte 00H,28H oder D8H annehmen. 00H heisst keine Y-Richtung 28H entspricht 40D also 40 Zeichen pro Bewegung mehr (eine Bildschirmzeile) und D8H entspricht -40D also eine Zeile nach oben.
Das dritte Byte dient als Zaehler fuer die Bewegungen eines Feindes nach unten. Nach 24 Bewegungen (da die erste Zeile nicht benutzt wird) ist ein Feind wieder in seiner Ausgangs-zeile.
Die X-Richtung ist fuer alle Feinde immer gleich.
Sobald ein Feind am linken oder rechten Bildschirmrand ist, wird die X-Richtung fuer alle umgekehrt.
Die Tabelle wird, da sie veraendert werden muss, zu Beginn jeder Runde von 7900H nach 7880H kopiert.
Zwei weitere Tabellen sind nach dem gleichen Prinzip aufgebaut.
Die Tabellen der Schuesse und die der Explosionen.
Die Schusstabelle besteht aus 16 Bytes, jeweils LSB und MSB der Schussposition im Bildschirmspeicher. Sie wird ab 7870H angelegt. Die Explosionstabelle besteht aus 24 Bytes ab 7850H. Sie hat zusaetzlich zu der Position der Explosion noch ein Byte als Zaehler fuer die drei Phasen der Explosion.
Die Anfangsgrafik beginnt bei 7A00H und belegt 1024 Byte.
Die definierbaren Zeichen beginnen bei 7E00 und belegen 512 Bytes. Sie sind zusaetzlich noch einmal vergroessert im Anschluss an das Listing ausgedruckt.
Die ersten 32 Zeichen von 128 bis 159 sind fuer 8 Feindtypen

in je zwei Bewegungszustaenden definiert.
Danach folgen die Zeichen fuer Schiff.Schuesse. Explosionen usw.
Doch nun wollen wir Sie nicht weiter vom tippen abhalten.
Wir wuenschen Ihnen viel Glueck beim Kampf gegen die Zalonen.
Das Programmlisting:

```
7000 21 00 7E 11 00 F4 01 00   A5
7008 02 ED B0 3E 01 CD B0 3F   9A
7010 CD F0 70 21 00 7A 11 00   D9
7018 44 01 00 04 ED B0 21 00   07
7020 F0 11 01 F0 01 FF 03 36   2B
7028 06 ED B0 CD F0 14 2A AA   48
7030 40 7C E6 01 C6 44 67 FE   12
7038 44 28 05 7D FE 18 30 EB   1F
7040 7E FE 7F 28 0A 36 7F 11   F3
7048 00 AC 19 36 02 18 10 E5   0A
7050 D1 EB 01 00 36 09 7E 12   8C
7058 EB 11 00 AC 19 36 05 CD   C9
7060 2B 00 FE 30 DA 2B 70 FE   CC
7068 3A 38 08 FE 53 C2 2B 70   28
7070 C3 80 70 32 3C 47 D6 30   6E
7078 32 00 42 C3 2B 70 00 00   D2
7080 21 00 00 22 02 42 3E 07   CC
7088 32 01 42 3A 3C 47 D6 30   38
7090 32 00 42 3E 07 D3 F8 3E   C2
7098 F7 D3 F9 21 E4 79 06 05   4C
70A0 36 B0 23 10 FB 3E 05 D3   2A
70A8 F8 3E 03 D3 F9 21 D0 07   FD
70B0 22 0C 42 C3 80 74 00 00   27
70B8 00 00 00 00 00 00 00 00   00
70C0 00 00 00 00 00 00 00 00   00
70C8 00 00 00 00 00 00 00 00   00
70D0 00 00 00 00 00 00 00 00   00
70D8 00 00 00 00 00 00 00 00   00
70E0 00 00 00 00 00 00 00 00   00
70E8 00 00 00 00 00 00 00 00   00
70F0 3E 07 D3 F8 3E FF D3 F9   19
70F8 CD B0 38 C9 00 00 00 00   7E
7100 DD 21 80 78 ED 4B 08 42   78
7108 DD 6E 00 DD 66 01 7C B5   C0
7110 CA B3 71 36 20 23 36 20   BD
7118 2B 09 DD 5E 02 16 00 7B   02
7120 FE D8 20 02 16 FF 19 B7   DD
7128 28 0C FE 28 28 05 DD 35   99
7130 03 18 03 DD 34 03 DD 7E   8D
7138 03 B7 20 04 DD 36 02 00   F3
7140 FE 18 20 08 DD 36 02 00   53
7148 DD 36 03 00 11 E8 47 DF   35
7150 38 04 11 40 FC 19 11 28   D8
7158 44 DF 30 04 11 C0 03 19   44
7160 DD 75 00 DD 74 01 3A 01   DF
7168 42 E6 07 07 07 C6 80 5F   E2
7170 3A 0A 42 E6 02 83 77 23   8B
7178 3C 77 11 FF AB 19 3A 01   C2
7180 42 3C 77 23 77 DD 7E 03   ED
7188 B7 20 13 CD F0 14 3A 00   F5
7190 42 5F 3A AA 40 BB 30 04   B4
7198 DD 36 02 28 18 15 DD 7E   C5
71A0 02 FE 28 20 0E CD F0 14   27
71A8 3A AA 40 FE 01 30 04 DD   34
71B0 36 02 D8 11 04 00 DD 19   1B
71B8 DD 7D B7 C2 04 71 C9 00   11
```

```
71C0 3E 08 D3 F8 DB F9 B7 28    C4
71C8 01 3D D3 F9 21 28 44 11    AB
71D0 28 00 06 16 7E FE 20 20    00
71D8 05 19 10 F8 18 06 21 01    66
71E0 00 22 08 42 21 4F 44 06    26
71E8 16 7E FE 20 20 04 19 10    FF
71F0 F8 C9 21 FF FF 22 08 42    4C
71F8 C9 00 00 00 00 00 00 00    C9
7200 3A 0A 42 3C 32 0A 42 CD    0D
7208 00 71 CD C0 71 CD 38 74    E8
7210 3A 40 F8 E6 20 28 01 2B    CC
7218 3A 40 F8 E6 40 28 01 23    E4
7220 7D FE 98 20 01 23 FE BC    11
7228 20 01 2B 22 06 42 CD 50    D3
7230 74 3A 40 F8 E6 80 28 03    77
7238 CD 70 72 CD 80 73 3A 40    E9
7240 F8 FE 06 CA 00 70 3A 0B    7B
7248 42 B7 C2 00 72 C3 80 74    E4
7250 00 00 00 00 00 00 00 00    00
7258 00 00 00 00 00 00 00 00    00
7260 00 00 00 00 00 00 00 00    00
7268 00 00 00 00 00 00 00 00    00
7270 21 70 78 7E 23 B6 28 07    8F
7278 23 7D FE 80 20 F5 C9 2B    27
7280 EB 2A 06 42 01 D9 FF 09    3F
7288 EB 73 23 72 EB 36 A6 11    CB
7290 00 AC 19 36 03 C9 00 00    C7
7298 E5 FD 21 50 78 FD 6E 00    36
72A0 FD 66 01 7C B5 28 12 FD    CC
72A8 23 FD 23 FD 23 FD 7D FE    DB
72B0 68 C2 9D 72 E1 21 00 00    3B
72B8 C9 E1 7E E6 01 28 01 2B    63
72C0 FD 75 00 FD 74 01 FD 36    17
72C8 02 00 EB FD 21 80 78 FD    00
72D0 6E 00 FD 66 01 DF 28 0E    E7
72D8 01 04 00 FD 09 FD 7D B7    3C
72E0 20 ED 21 00 00 C9 36 20    4D
72E8 23 36 20 21 00 00 FD 75    0C
72F0 00 FD 74 01 2A 02 42 3A    1A
72F8 00 42 07 3C 5F 16 00 19    13
7300 22 02 42 3A 0B 42 3D 32    5C
7308 0B 42 21 00 00 3E 08 D3    87
7310 F8 3E 0F D3 F9 C9 00 00    DA
7318 00 00 00 00 00 00 00 00    00
7320 3E 30 3C 09 38 FC 3D ED    11
7328 42 12 13 C9 D5 ED 5B 0C    59
7330 42 DF 38 1A EB 01 D0 07    36
7338 09 22 0C 42 EB 11 E4 79    D2
7340 1A FE B0 20 06 13 7B FE    7A
7348 EE 20 F5 3E B0 12 D1 01    D5
7350 F0 D8 CD 20 73 01 18 FC    3D
7358 CD 20 73 01 9C FF CD 20    E9
7360 73 01 F6 FF CD 20 73 01    CA
7368 FF FF CD 20 73 21 D8 79    D0
7370 11 00 44 01 28 00 ED B0    1B
7378 C9 00 00 00 00 00 00 00    C9
7380 DD 21 70 78 DD 6E 00 DD    0E
7388 66 01 7C B5 28 36 7E FE    72
7390 A0 38 1A 36 20 11 D8 FF    30
7398 19 11 28 44 DF 30 05 21    CB
73A0 00 00 18 14 7E FE 80 38    60
```

```
73A8 07 FE A0 30 03 CD 98 72    AF
73B0 3A 0A 42 E6 01 C6 A6 77    50
73B8 DD 75 00 DD 74 01 11 00    B5
73C0 AC 19 36 09 DD 23 DD 23    04
73C8 DD 7D FE 80 C2 84 73 DD    6E
73D0 21 50 78 DD 6E 00 DD 66    77
73D8 01 7C B5 28 28 DD 7E 02    DF
73E0 07 C6 AA FE B0 20 08 36    83
73E8 20 23 36 20 21 00 00 DD    97
73F0 75 00 DD 74 01 77 23 3C    9D
73F8 77 11 FF AB 19 D6 09 77    A1
7400 23 77 DD 34 02 DD 23 DD    BA
7408 23 DD 23 DD 7D FE 68 C2    A5
7410 D3 73 2A 02 42 11 DE 79    1C
7418 CD 2C 73 2A 02 42 ED 5B    22
7420 04 42 DF 38 09 22 04 42    CE
7428 11 F8 79 CD 4F 73 C9 00    DD
7430 00 00 00 00 00 00 00 00    00
7438 2A 06 42 06 03 7E FE A0    97
7440 38 09 36 20 23 10 F6 2B    EB
7448 2B 2B C9 F1 C3 30 75 00    78
7450 06 A0 3A 0A 42 E6 01 28    3B
7458 02 06 A3 78 77 23 3C 77    70
7460 23 3C 77 11 FE AB 19 36    DF
7468 06 23 36 02 23 36 06 C9    89
7470 00 00 00 00 00 00 00 00    00
7478 00 00 00 00 00 00 00 00    00
7480 21 00 44 11 01 44 01 FF    BB
7488 03 36 20 ED B0 21 D8 79    68
7490 11 00 44 01 28 00 ED B0    1B
7498 21 00 78 11 00 F0 01 28    C3
74A0 00 ED B0 3A 01 42 3C E6    3C
74A8 07 32 01 42 21 AB 47 22    B1
74B0 06 42 21 00 79 11 80 78    EB
74B8 01 80 00 ED B0 21 01 00    40
74C0 22 08 42 21 50 78 11 51    B7
74C8 78 01 2F 00 36 00 ED B0    7B
74D0 3E 20 32 08 42 3A 00 42    59
74D8 3C 32 00 42 21 28 78 11    82
74E0 3C 46 01 10 00 ED B0 2A    5A
74E8 06 42 CD 50 74 3E 18 32    61
74F0 89 71 06 0C C5 CD 00 71    0F
74F8 CD C0 71 21 3C F2 11 3D    9B
7500 F2 01 0F 00 ED 5F 77 ED    B2
7508 B0 01 00 18 CD 60 00 C1    B7
7510 10 E2 3E 20 32 89 71 21    9D
7518 3C 46 11 3D 46 01 0F 00    26
7520 77 ED B0 C3 00 72 00 00    49
7528 00 00 00 00 00 00 00 00    00
7530 06 1F 3E 07 D3 F8 3E E7    5A
7538 D3 F9 C5 3E 09 D3 F8 78    1B
7540 E6 0F D3 F9 3E 06 D3 F8    D0
7548 3E 1F 90 D3 F9 3A 0A 42    3F
7550 3C 32 0A 42 CD 00 71 CD    C5
7558 C0 71 CD 80 73 2A 50 78    E3
7560 7C B5 20 18 2A 06 42 ED    C8
7568 5F E6 01 28 01 23 22 50    04
7570 78 32 52 78 3E 08 D3 F8    85
7578 3E 0F D3 F9 C1 10 BB 21    C6
7580 EE 79 7E FE B0 28 09 2B    EF
7588 7D FE E3 20 F5 C3 A0 75    4B
7590 36 20 3A 01 42 3D 32 01    43
```

```
7598 42 C3 80 74 00 00 00 00   F9
75A0 21 FA 79 11 60 7D 01 06   89
75A8 00 ED B0 C3 00 70 00 00   D0
75B0 21 00 00 22 04 42 31 00   AA
75B8 60 C3 00 70 00 00 00 00   93
75C0 00 00 00 00 00 00 00 00   00
75C8 00 00 00 00 00 00 00 00   00
75D0 00 00 00 00 00 00 00 00   00
75D8 00 00 00 00 00 00 00 00   00
75E0 00 00 00 00 00 00 00 00   00
75E8 00 00 00 00 00 00 00 00   00
75F0 00 00 00 00 00 00 00 00   00
75F8 00 00 00 00 00 00 00 00   00
7600 00 00 00 00 00 00 00 00   00
7608 00 00 00 00 00 00 00 00   00
7610 00 00 00 00 00 00 00 00   00
7618 00 00 00 00 00 00 00 00   00
```

Nun folgen die Tabellen und Anfangsgrafik:

```
7800 02 02 02 02 02 03 03 03   13
7808 03 03 03 06 06 06 06 06   27
7810 06 06 06 06 06 06 06 02   2C
7818 02 02 02 02 02 02 02 02   10
7820 02 03 03 03 03 03 03 03   17
7828 2A 2A 20 47 45 54 20 52   C6
7830 45 41 44 59 20 2A 2A 20   B7
7838 00 00 00 00 00 00 00 00   00
7840 00 00 00 00 00 00 00 00   00
7848 00 00 00 00 00 00 00 00   00
7850 00 00 00 00 00 00 00 00   00
7858 00 00 00 00 00 00 00 00   00
7860 00 00 00 00 00 00 00 00   00
7868 00 00 00 00 00 00 00 00   00
7870 00 00 00 00 00 00 00 00   00
7878 00 00 00 00 00 00 00 00   00
7880 00 00 00 00 00 00 00 00   00
7888 00 00 00 00 00 00 00 00   00
7890 00 00 00 00 00 00 00 00   00
7898 00 00 00 00 00 00 00 00   00
78A0 00 00 00 00 00 00 00 00   00
78A8 00 00 00 00 00 00 00 00   00
78B0 00 00 00 00 00 00 00 00   00
78B8 00 00 00 00 00 00 00 00   00
78C0 00 00 00 00 00 00 00 00   00
78C8 00 00 00 00 00 00 00 00   00
78D0 00 00 00 00 00 00 00 00   00
78D8 00 00 00 00 00 00 00 00   00
78E0 00 00 00 00 00 00 00 00   00
78E8 00 00 00 00 00 00 00 00   00
78F0 00 00 00 00 00 00 00 00   00
78F8 00 00 00 00 00 00 00 00   00
7900 50 44 00 00 54 44 00 00   2C
7908 58 44 00 00 5C 44 00 00   3C
7910 60 44 00 00 64 44 00 00   4C
7918 68 44 00 00 6C 44 00 00   5C
7920 A2 44 00 00 A6 44 00 00   D0
7928 AA 44 00 00 AE 44 00 00   E0
7930 B2 44 00 00 B6 44 00 00   F0
7938 BA 44 00 00 F4 44 00 00   36
7940 F8 44 00 00 FC 44 00 00   7C
```

```
7948 00 45 00 00 04 45 00 00   8E
7950 08 45 00 00 46 45 00 00   D8
7958 4A 45 00 00 4E 45 00 00   22
7960 52 45 00 00 56 45 00 00   32
7968 98 45 00 00 9C 45 00 00   BE
7970 A0 45 00 00 A4 45 00 00   CE
7978 EA 45 00 00 F2 45 00 00   66
7980 00 00 00 00 00 00 00 00   00
7988 00 00 00 00 00 00 00 00   00
7990 00 00 00 00 00 00 00 00   00
7998 00 00 00 00 00 00 00 00   00
79A0 00 00 00 00 00 00 00 00   00
79A8 00 00 00 00 00 00 00 00   00
79B0 00 00 00 00 00 00 00 00   00
79B8 00 00 00 00 00 00 00 00   00
79C0 00 00 00 00 00 00 00 00   00
79C8 00 00 00 00 00 00 00 00   00
79D0 00 00 00 00 00 00 00 00   00
79D8 B6 B7 B8 B9 BA 20 30 30   18
79E0 32 31 32 20 20 20 20 20   35
79E8 20 20 20 20 20 20 20 B1   91
79F0 B2 B3 B4 B5 B6 B7 B8 B9   AC
79F8 BA 20 20 30 36 37 33 30   FA
7A00 20 20 CA CA CA CA CA 20   52
7A08 CD CA CA CA E0 20 CA 20   15
7A10 20 20 20 20 CD CA CA CA   AB
7A18 E0 20 CD CA CA CA E0 20   2B
7A20 CD CA CA CA E0 20 20 20   6B
7A28 20 20 20 20 20 CD E2 20   6F
7A30 CA 20 20 20 CA 20 CA 20   FE
7A38 20 20 20 20 CA 20 20 20   AA
7A40 CA 20 CA 20 20 20 D1 20   05
7A48 CA 20 20 20 CA 20 20 20   54
7A50 20 20 20 20 CD E2 20 20   6F
7A58 CA 20 20 20 CA 20 CA 20   FE
7A60 20 20 20 20 CA 20 20 20   AA
7A68 CA 20 CA 20 20 20 20 20   54
7A70 CA 20 20 20 CA 20 20 20   54
7A78 20 20 20 CD E2 20 20 20   6F
7A80 CA CA CA CA CA 20 CA 20   FC
7A88 20 20 20 20 CA CA CA CA   A8
7A90 CA 20 CA 20 CA CA CA 20   52
7A98 CA CA CA CA CA 20 20 20   52
7AA0 20 20 CD E2 20 20 20 20   6F
7AA8 CA 20 20 20 CA 20 CA 20   FE
7AB0 20 20 20 20 CA 20 20 20   AA
7AB8 CA 20 CA 20 20 20 CA 20   FE
7AC0 CA 20 20 20 CA 20 20 20   54
7AC8 20 20 CA 20 20 20 20 20   AA
7AD0 CA 20 20 20 CA 20 CA 20   FE
7AD8 20 20 20 20 CA 20 20 20   AA
7AE0 CA 20 CA 20 20 20 CA 20   FE
7AE8 CA 20 20 20 CA 20 20 20   54
7AF0 20 20 CA CA CA CA CA 20   52
7AF8 CA 20 20 20 CA 20 CA CA   A8
7B00 CA CA CA 20 CA 20 20 20   A8
7B08 CA 20 E4 CA CA CA E2 20   2E
7B10 CA 20 20 20 CA 20 20 20   54
7B18 20 20 20 20 20 20 20 20   00
7B20 20 20 20 20 20 20 20 20   00
7B28 20 20 20 20 20 20 20 20   00
```

```
7B30 20 20 20 20 20 20 20 20   00
7B38 20 20 20 20 20 20 20 20   00
7B40 20 20 44 69 65 20 5A 61   2D
7B48 6C 6F 6E 65 6E 20 61 75   12
7B50 73 20 64 65 72 20 47 61   96
7B58 6C 61 78 69 65 20 4E 47   C8
7B60 43 2D 35 34 37 20 20 20   70
7B68 20 20 67 72 65 69 66 65   B2
7B70 6E 20 64 69 65 20 45 72   97
7B78 64 65 20 6D 69 74 20 69   BC
7B80 68 72 65 72 20 67 65 73   10
7B88 61 6D 74 65 6E 20 20 20   75
7B90 20 20 46 6C 6F 74 74 65   AE
7B98 20 61 6E 2E 20 53 69 65   5E
7BA0 20 73 69 6E 64 20 6D 69   C4
7BA8 74 20 64 65 6D 20 6C 65   8B
7BB0 74 7A 74 65 6E 20 20 20   95
7BB8 20 20 46 6C 61 67 67 73   94
7BC0 63 68 69 66 66 20 64 65   E9
7BC8 72 20 52 61 75 6D 2D 42   96
7BD0 61 73 69 73 20 48 61 6E   E7
7BD8 67 65 6C 61 72 20 20 20   6B
7BE0 20 20 67 65 66 6C 6F 68   B5
7BE8 65 6E 20 75 6E 64 20 6D   C7
7BF0 75 65 73 73 65 6E 20 73   26
7BF8 69 63 68 20 64 75 72 63   02
7C00 68 20 64 69 65 20 20 20   1A
7C08 20 20 20 20 20 66 65 69   D4
7C10 6E 64 6C 69 63 68 65 6E   45
7C18 20 52 65 69 68 65 6E 20   9B
7C20 6B 61 65 6D 70 66 65 6E   47
7C28 2E 20 20 20 20 20 20 20   0E
7C30 20 20 20 20 20 20 20 20   00
7C38 20 20 20 20 20 20 20 20   00
7C40 20 20 20 20 20 20 20 20   00
7C48 20 20 20 20 20 20 20 20   00
7C50 20 20 20 20 20 20 20 20   00
7C58 20 20 44 69 65 20 53 74   39
7C60 65 75 65 72 75 6E 67 20   1B
7C68 64 65 73 20 53 63 68 69   E3
7C70 66 66 73 20 67 65 73 63   01
7C78 68 69 65 68 74 20 20 20   72
7C80 20 20 6D 69 74 20 64 65   73
7C88 6E 20 50 66 65 69 6C 74   F2
7C90 61 73 74 65 6E 2E 20 49   B2
7C98 68 72 20 4C 61 73 65 72   F1
7CA0 20 77 69 72 64 20 20 20   36
7CA8 20 20 20 20 20 6D 69 74   EA
7CB0 20 64 65 72 20 4C 65 65   91
7CB8 72 74 61 73 74 65 20 61   14
7CC0 75 73 67 65 6C 6F 65 73   67
7CC8 74 2E 20 20 20 20 20 20   62
7CD0 20 20 20 20 20 20 20 20   00
7CD8 20 20 20 20 20 20 20 20   00
7CE0 20 20 20 20 20 20 20 20   00
7CE8 20 20 20 20 20 20 20 20   00
7CF0 20 20 20 20 20 20 20 20   00
7CF8 20 20 0F D3 D3 D3 D3 D3   3E
7D00 D3 D3 D3 D3 D3 D3 D3 D3   98
7D08 D3 D3 D3 D3 D3 D3 D3 D3   98
7D10 D3 D3 D3 D3 D3 D3 D3 D3   98
7D18 D3 D3 D3 D3 E3 20 20 20   8F
```

```
7D20 20 20 CF 47 45 53 43 48   79
7D28 57 49 4E 44 49 47 48 45   52
7D30 49 54 20 28 30 2E 2E 2E   9F
7D38 39 29 20 20 35 20 20 20   37
7D40 20 20 20 20 DB 20 20 20   BB
7D48 20 20 CF 42 45 53 54 45   82
7D50 53 20 45 52 47 45 42 4E   26
7D58 49 53 20 20 20 20 20 20   5C
7D60 20 30 30 30 30 30 20 20   50
7D68 20 20 20 20 DB 20 20 20   BB
7D70 20 20 CF 44 72 75 65 63   02
7D78 6B 65 6E 20 53 69 65 20   9F
7D80 3C 53 3E 20 75 6D 20 7A   69
7D88 75 20 62 65 67 69 6E 6E   08
7D90 65 6E 20 21 DB 20 20 20   4F
7D98 20 20 E3 DA DA DA DA DA   65
7DA0 DA DA DA DA DA DA DA DA   D0
7DA8 DA DA DA DA DA DA DA DA   D0
7DB0 DA DA DA DA DA DA DA DA   D0
7DB8 DA DA DA DA DF 20 20 20   A7
7DC0 20 20 20 20 20 20 20 20   00
7DC8 20 20 20 20 20 20 20 20   00
7DD0 20 20 20 20 20 20 20 20   00
7DD8 20 20 20 20 20 20 20 20   00
7DE0 20 20 20 20 20 20 20 20   00
7DE8 00 00 00 00 00 00 00 00   00
7DF0 00 00 00 00 00 00 00 00   00
7DF8 00 00 00 00 00 00 00 00   00
7E00 02 02 07 0E 1C 3F 0C 18   98
7E08 40 40 E0 70 38 FC 30 18   4C
7E10 04 02 07 0E 1C 3F 0C 06   88
7E18 20 40 E0 70 38 FC 30 60   74
7E20 0F 19 39 1F 0F 0A 14 0A   B7
7E28 F0 98 9C F8 F0 50 28 50   D4
7E30 0F 19 39 1F 0F 0A 14 28   D5
7E38 F0 98 9C F8 F0 50 28 14   98
7E40 07 1F 29 29 1F 0F 03 01   AA
7E48 E0 F8 94 94 F8 F0 C0 80   28
7E50 07 1F 32 32 1F 0F 03 01   BC
7E58 E0 F8 4C 4C F8 F0 C0 80   98
7E60 01 02 04 F9 7F 2B 01 00   AB
7E68 80 40 20 9F FE D4 80 00   D1
7E70 01 03 04 09 3F 6B A1 00   5C
7E78 80 C0 20 90 FC D6 85 00   47
7E80 00 82 42 31 2E 4A 11 00   7E
7E88 00 41 42 8C 74 52 88 00   5D
7E90 00 02 02 01 FE 2A 09 00   36
7E98 00 40 40 80 7F 54 90 00   63
7EA0 00 01 05 04 08 08 05 05   24
7EA8 50 50 08 08 10 50 40 00   50
7EB0 05 05 08 08 04 05 01 00   24
7EB8 00 40 50 10 08 08 50 50   50
7EC0 00 0D 13 23 45 48 30 00   00
7EC8 00 B0 C8 C4 A2 12 0C 00   FC
7ED0 00 7D 83 83 7D 00 00 00   00
7ED8 00 BE C1 C1 BE 00 00 00   FE
7EE0 00 3C 24 27 27 24 3C 00   0E
7EE8 00 3C 24 E4 E4 24 3C 00   88
7EF0 00 0F 09 39 39 09 0F 00   A2
7EF8 00 F0 90 9C 9C 90 F0 00   38
7F00 00 04 0E 1B 1B 0E 04 00   5A
```

7F08	18	18	3C	DB	99	7E	7E	24	00
7F10	00	20	70	D8	D8	70	20	00	D0
7F18	00	04	0E	15	15	0E	04	00	4E
7F20	18	18	3C	E7	C3	7E	7E	24	36
7F28	00	20	70	A8	A8	70	20	00	70
7F30	10	08	10	08	10	00	00	00	40
7F38	00	00	00	08	10	08	10	08	38
7F40	18	18	18	18	00	00	00	00	60
7F48	00	00	00	00	18	18	18	18	60
7F50	11	0C	03	02	0F	10	01	06	48
7F58	08	B0	40	FC	B2	E0	18	04	A2
7F60	01	14	01	55	02	24	01	08	9A
7F68	10	80	24	40	AA	80	28	80	C6
7F70	00	00	04	00	12	00	04	00	1A
7F78	00	90	00	40	00	08	40	00	18
7F80	10	10	D6	FE	D6	10	00	00	DA
7F88	04	04	4C	54	64	44	40	40	D0
7F90	08	10	30	10	10	18	10	20	B0
7F98	18	20	40	48	58	48	50	20	D0
7FA0	04	04	4C	54	64	44	40	40	D0
7FA8	00	00	00	08	10	20	00	00	38
7FB0	18	20	40	58	34	04	08	30	40
7FB8	18	20	40	40	40	44	48	30	B4
7FC0	18	24	44	44	44	44	48	30	C4
7FC8	18	24	44	48	50	68	44	40	04
7FD0	0C	10	20	28	30	24	28	30	10
7FD8	FF	00	E3	14	14	14	14	14	46
7FE0	C7	08	93	24	48	4F	40	3F	9C
7FE8	F3	04	E4	02	01	BF	00	BF	5C
7FF0	FC	00	F8	40	20	A0	20	C0	D4
7FF8	00	00	00	00	00	00	00	00	00

```
7000  21007E   LD    HL,7E00H      ; Zeichensatz von 7E00H
7003  1100F4   LD    DE,0F400H     ; nach F400H in den Zeichenspeicher
7006  010002   LD    BC,0200H      ; verschieben.
7009  EDB0     LDIR                ;
700B  3E01     LD    A,01H         ; CHAR 2 einschalten. Argument
700D  CDB03F   CALL  3FB0H         ; im Akku ist um eins vermindert.
7010  CDF070   CALL  70F0H         ; Unterprg.: Soundoff und LGR
7013  21007A   LD    HL,7A00H      ; Adresse der Anfangsgrafik
7016  110044   LD    DE,4400H      ; Adresse Bildschirmanfang
7019  010004   LD    BC,0400H      ; Anzahl der Bytes
701C  EDB0     LDIR                ; Verschieben.
701E  2100F0   LD    HL,0F000H     ; Farbspeicher
7021  1101F0   LD    DE,0F001H     ; Farbspeicher +1
7024  01FF03   LD    BC,03FFH      ; Laenge -1
7027  3606     LD    (HL),06H      ; erstes Byte mit 06H belegen
7029  EDB0     LDIR                ; 1. Byte auf 2. usw. schieben.
702B  CDF014   CALL  14F0H         ; Unterprg.: Zufallszahl erzeugen.
702E  2AAA40   LD    HL,(40AAH)    ; Zahl in HL holen.
7031  7C       LD    A,H           ; MSB in Akku
7032  E601     AND   01H           ; Akku wird 00H oder 01H
7034  C644     ADD   A,44H         ; Akku wird 44H oder 45H
7036  67       LD    H,A           ; Zurueck in MSB. (HL = 4400H- 45FFH)
7037  FE44     CP    44H           ; Wenn MSB = 44H,
7039  2805     JR    Z,$+07H       ; dann nach 7040H
703B  7D       LD    A,L           ; LSB in Akku
703C  FE18     CP    18H           ; Wenn )= 18H
703E  30EB     JR    NC,$-13H      ; Dann neue Zahl erzeugen.
7040  7E       LD    A,(HL)        ; HL Zeiger auf Bildschirmadresse
7041  FE7F     CP    7FH           ; Zeichen auf HL = 7FH ?
7043  280A     JR    Z,$+0CH       ; Wenn ja dann nach 704FH
7045  367F     LD    (HL),7FH      ; sonst 7FH auf HL abspeichern
7047  1100AC   LD    DE,0AC00H     ; Offset zum Farbspeicher
704A  19       ADD   HL,DE         ; zu HL addieren
704B  3602     LD    (HL),02H      ; 02H auf HL abspeichern (rot)
704D  1810     JR    $+12H         ; Weiter bei 705FH
704F  E5       PUSH  HL            ; HL auf den Stack
7050  D1       POP   DE            ; DE zurueck
7051  EB       EX    DE,HL         ; vertauschen
7052  010036   LD    BC,3600H      ; Offset zur Anfangsgrafik
7055  09       ADD   HL,BC         ; zu HL addieren
7056  7E       LD    A,(HL)        ; Originalzeichen der Anfangsgrafik
7057  12       LD    (DE),A        ; in den Bildschirmspeicher speichern
7058  EB       EX    DE,HL         ; Bildschirmadresse zurueck nach HL
7059  1100AC   LD    DE,0AC00H     ; Offset zum Farbspeicher
705C  19       ADD   HL,DE         ; zu HL addieren
705D  3605     LD    (HL),05H      ; 02H auf HL abspeichern (gruen)
705F  CD2B00   CALL  002BH         ; Unterprg.: GETCHR (Tastaturabfrage)
7062  FE30     CP    30H           ; < 30H  ASC("0")
7064  DA2B70   JP    C,702BH       ; Wenn ja, dann Sprung zur Schleife
7067  FE3A     CP    3AH           ; < 3AH  ASC(":")
7069  3808     JR    C,$+0AH       ; Wenn ja, dann zu 7073H
706B  FE53     CP    53H           ; = 53H  ASC("5")
706D  C22B70   JP    NZ,702BH      ; Wenn nicht, dann zur Schleife
7070  C38070   JP    7080H         ; Sonst Spiel starten
7073  323C47   LD    (473CH),A     ; gedrueckte Zahl im Bildschirm anzeigen
7076  D630     SUB   30H           ; - 30H ergibt einen Wert von 00H-09H
7078  320042   LD    (4200H),A     ; in Variable fuer Level abspeichern
707B  C32B70   JP    702BH         ; und zurueck zur Schleife.
```

; *** STARTEN DES SPIELS ***

```
7080  210000    LD    HL,0000H     ; HL [illegible] mit Startpunktzahl belegen
7083  220242    LD    (4202H),HL   ; in Variable fuer Punktzahl abspeichern
7086  3E07      LD    A,07H        ; Zaehler fuer Feindtyp auf 7 setzten
7088  320142    LD    (4201H),A    ; und abspeichern.
708B  3A3C47    LD    A,(473CH)    ; Level aus Bildschirmspeicher holen
708E  D630      SUB   30H          ; - 30H
7090  320042    LD    (4200H),A    ; in Variable fuer Level abspeichern
7093  3E07      LD    A,07H        ; 07H          (SOUND 7,247 :
7095  D3F8      OUT   (0F8H),A     ; auf Port F8H   schaltet Rauschen
7097  3EF7      LD    A,0F7H       ; F7H          auf Kanal 1 ein)
7099  D3F9      OUT   (0F9H),A     ; auf Port F9H
709B  21E479    LD    HL,79E4H     ; Auf die Adresse fuer die Anzeige der
709E  0605      LD    B,05H        ; restlichen Schiffe, 5 mal
70A0  36B0      LD    (HL),0B0H    ; 0B0H (kleines Schiff) abspeichern
70A2  23        INC   HL           ;
70A3  10FB      DJNZ  $-03H        ;
70A5  3E05      LD    A,05H        ; 05H
70A7  D3F8      OUT   (0F8H),A     ; auf Port F8H   (SOUND 5,3   :
70A9  3E03      LD    A,03H        ; 03H          Frequenz des dritten
70AB  D3F9      OUT   (0F9H),A     ; auf Port F9H   Kanals vorbelegen)
70AD  21D007    LD    HL,07D0H     ; 07D0H = 2000D als erste Grenze fuer
70B0  220C42    LD    (420CH),HL   ; Bonusschiff abspeichern
70B3  C38074    JP    7480H        ; Sprung zum Bildaufbau
```

; *** UNTERPROGRAMM SOUNDOFF UND LGR ***

```
70F0  3E07      LD    A,07H        ; 07H          (SOUND 7,255 :
70F2  D3F8      OUT   (0F8H),A     ; auf Port F8H   alle Kanaele
70F4  3EFF      LD    A,0FFH       ; 0FFH         abschalten)
70F6  D3F9      OUT   (0F9H),A     ; auf Port F9H
70F8  CDB038    CALL  38B0H        ; Rom-Routine fuer LGR
70FB  C9        RET                ;
```

; *** UNTERPROGRAMM FUER BEWEGUNG DER FEINDE ***

```
7100  DD218078  LD    IX,7880H     ; IX auf Tabelle der Feindpositionen
7104  ED4B0842  LD    BC,(4208H)   ; BC mit X-Bewegungsrichtung vorbelegen
7108  DD6E00    LD    L,(IX+00H)   ; L ist LSB der Position
710B  DD6601    LD    H,(IX+01H)   ; H ist MSB der    "
710E  7C        LD    A,H          ; MSB
710F  B5        OR    L            ; oderiert mit LSB = 0 ?
7110  CAB371    JP    Z,71B3H      ; Wenn ja, dann existiert Feind nicht
7113  3620      LD    (HL),20H     ; Erstes Byte mit 20H (SPACE) loeschen
7115  23        INC   HL           ; und
7116  3620      LD    (HL),20H     ; zweites Byte ebenso
```

```
7118  2B        DEC   HL            ; HL zurueck auf erstes Byte
7119  09        ADD   HL,BC         ; X-Richtung addieren
711A  DD5E02    LD    E,(IX+02H)    ; E ist LSB der Y-Richtung
711D  1600      LD    D,00H         ; D loeschen
711F  7B        LD    A,E           ; LSB in Akku
7120  FED8      CP    0D8H          ; = 0D8H ? (= -40D)
7122  2002      JR    NZ,$+04H      ; nein, es ist 00H oder 28H (= 40D)
7124  16FF      LD    D,0FFH        ; sonst MSB = 0FFH fuer Addition
7126  19        ADD   HL,DE         ; Y-Richtung zur Position addieren
7127  B7        OR    A             ; Y-Richtung = 0 ?
7128  280C      JR    Z,$+0EH       ; wenn ja, dann nach 7136H
712A  FE28      CP    28H           ; = +40D ?
712C  2805      JR    Z,$+07H       ; wenn ja, dann nach 7133H
712E  DD3503    DEC   (IX+03H)      ; sonst Y-Bewegungszaehler -1
7131  1803      JR    $+05H         ; und nach 7136H
7133  DD3403    INC   (IX+03H)      ; Y-Bewegungszaehler +1
7136  DD7E03    LD    A,(IX+03H)    ; Y-Bewegungszaehler in Akku
7139  B7        OR    A             ; Flags setzen
713A  2004      JR    NZ,$+06H      ; () 0 ? dann nach 7140H
713C  DD360200  LD    (IX+02H),00H  ; sonst LSB der Y-Richtung loeschen
7140  FE18      CP    18H           ; = 18H
7142  2008      JR    NZ,$+0AH      ; wenn nicht, dann nach 714CH
7144  DD360200  LD    (IX+02H),00H  ; sonst LSB der Y-Richtung
7148  DD360300  LD    (IX+03H),00H  ; und Y-Bewegungszaehler loeschen
714C  11E847    LD    DE,47E8H      ; Adresse von Bildschirmende +1
714F  DF        RST   18H           ; mit HL vergleichen (Position)
7150  3804      JR    C,$+06H       ; wenn HL ( DE dann nach 7156H
7152  1140FC    LD    DE,0FC40H     ; sonst 0FC40H (= -960D) zur Position
7155  19        ADD   HL,DE         ; addieren. (HL= zweite Bildschirmzeile
7156  112844    LD    DE,4428H      ; Adresse der 2. Bildschirmzeile in DE
7159  DF        RST   18H           ; mit HL vergleichen
715A  3004      JR    NC,$+06H      ; wenn HL )= DE dann nach 7160H
715C  11C003    LD    DE,03C0H      ; sonst 03C0H (= 960D) zur Position
715F  19        ADD   HL,DE         ; addieren. (HL= letzte Bildschirmzeile)
7160  DD7500    LD    (IX+00H),L    ; LSB der Position zurueck in Tabelle
7163  DD7401    LD    (IX+01H),H    ; MSB ebenso
7166  3A0142    LD    A,(4201H)     ; Feindtyp in Akku
7169  E607      AND   07H           ; nur von 00H bis 07H zugelassen
716B  07        RLCA                ; 2 mal links rotieren
716C  07        RLCA                ; d.h. mit 4 multiplizieren
716D  C680      ADD   A,80H         ; + 80H ergibt erstes Zeichen des Feinds
716F  5F        LD    E,A           ; in E zwischenspeichern
7170  3A0A42    LD    A,(420AH)     ; Zaehler in den Akku
7173  E602      AND   02H           ; mit 02H undieren. (gibt 00H oder 02H)
7175  83        ADD   A,E           ; E addieren, ergibt Phase des Feinds
7176  77        LD    (HL),A        ; auf Position abspeichern
7177  23        INC   HL            ; naechste Position
7178  3C        INC   A             ; naechstes Zeichen
7179  77        LD    (HL),A        ; abspeichern. Feind steht im Bildschirm
717A  11FFAB    LD    DE,0ABFFH     ; Offset zum Farbspeicher -1 (wegen INC)
717D  19        ADD   HL,DE         ; zur Position addieren
717E  3A0142    LD    A,(4201H)     ; Feindtyp
7181  3C        INC   A             ; +1 als Farbcode
7182  77        LD    (HL),A        ; in den Farbspeicher ablegen
7183  23        INC   HL            ; und
7184  77        LD    (HL),A        ; auf der naechsten Adresse
7185  DD7E03    LD    A,(IX+03H)    ; Y-Bewegungszaehler in Akku
7188  B7        OR    A             ; Flags setzen
7189  2013      JR    NZ,$+15H      ; () 0 ? dann nach 719EH
718B  CDF014    CALL  14F0H         ; Zufallszahl erzeugen
718E  3A0042    LD    A,(4200H)     ; Level in Akku
7191  5F        LD    E,A           ; in E zwischenspeichern
```

```
7192   3AAA40     LD     A,(40AAH)        ; Zufallszahl in Akku
7195   BB         CP     E                ; mit Level vergleichen
7196   3004       JR     NC,$+06H         ; wenn Zahl > Level dann nach 719CH
7198   DD360228   LD     (IX+02H),28H     ; 28H (= +40D) in Y-Richtung abspeichern
719C   1815       JR     $+17H            ; und nach 71B3H
719E   DD7E02     LD     A,(IX+02H)       ; Y-Richtung in den Akku
71A1   FE28       CP     28H              ; = 28H (= +40D)
71A3   200E       JR     NZ,$+10H         ; wenn nicht, dann nach 71B3H
71A5   CDF014     CALL   14F0H            ; Zufallszahl erzeugen
71A8   3AAA40     LD     A,(40AAH)        ; Zahl in Akku
71AB   FE01       CP     01H              ; < 01H  (koennen Sie aendern)
71AD   3004       JR     NC,$+06H         ; wenn nicht, dann nach 71B3H
71AF   DD3602D8   LD     (IX+02H),0D8H    ; sonst 0D8H (= -40D) in Y-Richtung
71B3   110400     LD     DE,0004H         ; 0004H fuer 4 Tabellenplaetze
71B6   DD19       ADD    IX,DE            ; zu IX addieren
71B8   DD         DEFB   0DDH             ; DD7D ist ein Sonderbefehl, der das
71B9   7D         LD     A,L              ; LSB von IX in den Akku laedt.
71BA   B7         OR     A                ; gleich 0 ? (bei IX = 7900H)
71BB   C20471     JP     NZ,7104H         ; nein ? dann naechsten Feind bewegen
71BE   C9         RET                     ; sonst zurueck
```

; *** UNTERPROGRAMM ZUR AENDERUNG DER X-RICHTUNG ***

```
71C0   3E08       LD     A,08H            ; 08H
71C2   D3F8       OUT    (0F8H),A         ; auf Port F8H
71C4   DBF9       IN     A,(0F9H)         ; Port F9H in den Akku (Lautstaerke 1)
71C6   B7         OR     A                ; Flags setzen
71C7   2801       JR     Z,$+03H          , schon 0 ? dann nach 71CAH
71C9   3D         DEC    A                ; sonst -1
71CA   D3F9       OUT    (0F9H),A         ; und auf Port F9H wieder ausgeben.
71CC   212844     LD     HL,4428H         ; HL auf 1. Byte, 2. Bildschirmzeile
71CF   112800     LD     DE,0028H         ; DE = 28H (= +40D)
71D2   0616       LD     B,16H            ; B = 16H (= 20D) als Zaehler der Zeilen
71D4   7E         LD     A,(HL)           ; Zeichen aus dem Bildschirm in Akku
71D5   FE20       CP     20H              ; = 20H (SPACE) ?
71D7   2005       JR     NZ,$+07H         ; wenn nicht, dann nach 71DEH
71D9   19         ADD    HL,DE            ; sonst HL = HL + 40D (naechste Zeile)
71DA   10F8       DJNZ   $-06H            ; Zaehler -1 nicht 0 ? dann nach 71D4H
71DC   1806       JR     $+08H            ; sonst nach 71E4H
71DE   210100     LD     HL,0001H         ; +1 (= nach rechts)
71E1   220842     LD     (4208H),HL       ; in Variable fuer X-Richtung
71E4   214F44     LD     HL,444FH         ; HL auf 39. Byte in 2. Bildschirmzeile
71E7   0616       LD     B,16H            ; B = 20D
71E9   7E         LD     A,(HL)           ; Zeichen in Akku
71EA   FE20       CP     20H              ; = 20H (SPACE) ?
71EC   2004       JR     NZ,$+06H         ; wenn nicht, dann nach 71F2H
71EE   19         ADD    HL,DE            ; HL = HL + 40D
71EF   10F8       DJNZ   $-06H            ; naechstes Zeichen testen
71F1   C9         RET                     ; und zurueck
71F2   21FFFF     LD     HL,0FFFFH        ; -1 (= nach links)
71F5   220842     LD     (4208H),HL       ; in Variable fuer X-Richtung
71F8   C9         RET                     ; und zurueck
```

; *** HAUPTSCHLEIFE ***

```
7200  3A0A42   LD    A,(420AH)     ; Zaehler in Akku
7203  3C       INC   A             ; +1
7204  320A42   LD    (420AH),A     ; wieder abspeichern
7207  CD0071   CALL  7100H         ; Unterprg.: Feindbewegung
720A  CDC071   CALL  71C0H         ; Unterprg.: X-Richtung
720D  CD3874   CALL  7438H         ; Unterprg.: Schiff loeschen
7210  3A40F8   LD    A,(0F840H)    ; Tastaturspeicher in Akku
7213  E620     AND   20H           ; Bit 5 isolieren (Pfeil nach links)
7215  2801     JR    Z,$+03H       ; nicht gesetzt ? dann nach 7218H
7217  2B       DEC   HL            ; sonst Schiffs-Position -1
7218  3A40F8   LD    A,(0F840H)    ; Tastaturspeicher in Akku
721B  E640     AND   40H           ; Bit 6 (Pfeil nach rechts)
721D  2801     JR    Z,$+03H       ; nicht gesetzt ? dann nach 7220H
721F  23       INC   HL            ; sonst Schiffs-Position +1
7220  7D       LD    A,L           ; LSB der Position in Akku
7221  FE98     CP    98H           ; = 98H ? (linker Rand)
7223  2001     JR    NZ,$+03H      ; wenn nicht, nach 7226H
7225  23       INC   HL            ; sonst Position +1
7226  FEBC     CP    0BCH          ; = BCH ? (rechter Rand -4)
7228  2001     JR    NZ,$+03H      ; wenn nicht, nach 722BH
722A  2B       DEC   HL            ; sonst Position -1
722B  220642   LD    (4206H),HL    ; Position wieder abspeichern
722E  CD5074   CALL  7450H         ; Unterprg.: Schiff und Farbe setzen
7231  3A40F8   LD    A,(0F840H)    ; Tastaturspeicher in Akku
7234  E680     AND   80H           ; Bit 7
7236  2803     JR    Z,$+05H       ; nicht gesetzt ? dann nach 723BH
7238  CD7072   CALL  7270H         ; Unterprg.: Schuss initialisieren
723B  CD8073   CALL  7380H         ; Unterprg.: Schuesse und Explosionen
723E  3A40F8   LD    A,(0F840H)    ; Tastaturspeicher in Akku
7241  FE06     CP    06H           ; = 06H  (BREAK & CLEAR) ?
7243  CA0070   JP    Z,7000H       ; wenn ja, zum Programmstart
7246  3A0B42   LD    A,(420BH)     ; Anzahl der verbliebenen Feinde
7249  B7       OR    A             ; Flags setzen
724A  C20072   JP    NZ,7200H      ; wenn <> 0 dann zur Hauptschleife
724D  C38074   JP    7480H         ; sonst naechster Level
```

; *** SCHUSS INITIALISIEREN ***

```
7270  217078   LD    HL,7870H      ; Anfang der Schusspositionstabelle
7273  7E       LD    A,(HL)        ; LSB in Akku
7274  23       INC   HL            ;
7275  B6       OR    (HL)          ; oderiert mit MSB
7276  2807     JR    Z,$+09H       ; = 0 ? dann nach 727FH
7278  23       INC   HL            ; naechste Tabellenposition
7279  7D       LD    A,L           ; LSB des Zeigers in Akku
727A  FE80     CP    80H           ; = 80H ? (HL = 7880H)
727C  20F5     JR    NZ,$-09H      ; wenn nicht, dann nach 7273H
```

```
727E  C9          RET                  ; zurueck, alle Schuesse belegt
727F  2B          DEC   HL             ; HL wieder auf LSB
7280  EB          EX    DE,HL          ; in DE
7281  2A0642      LD    HL,(4206H)     ; Schiffsposition
7284  01D9FF      LD    BC,0FFD9H      ; FFD9H (= -39D)
7287  09          ADD   HL,BC          ; addieren. Position ueber dem Schiff
7288  EB          EX    DE,HL          ; wieder vertauschen
7289  73          LD    (HL),E         ; LSB in Tabelle
728A  23          INC   HL             ; naechste Adresse
728B  72          LD    (HL),D         ; MSB abspeichern
728C  EB          EX    DE,HL          ; Position wieder in HL
728D  36A6        LD    (HL),0A6H      ; A6H ist def. Zeichen fuer Schuss
728F  1100AC      LD    DE,0AC00H      ; Offset zum Farbspeicher
7292  19          ADD   HL,DE          ; addieren
7293  3603        LD    (HL),03H       ; und mit 3 belegen
7295  C9          RET                  ; zurueck
```

; *** EXPLOSION INITIALISIEREN UND FEIND LOESCHEN ***

```
7298  E5          PUSH  HL             ; Trefferposition retten
7299  FD215078    LD    IY,7850H       ; IY auf Tabelle der Explosionen
729D  FD6E00      LD    L,(IY+00H)     ; LSB in L
72A0  FD6601      LD    H,(IY+01H)     ; MSB in H
72A3  7C          LD    A,H            ; MSB
72A4  B5          OR    L              ; oderiert mit LSB
72A5  2812        JR    Z,$+14H        ; = 0 ? dann nach 72B9H
72A7  FD23        INC   IY             ; sonst drei Tabellenplaetze weiter
72A9  FD23        INC   IY             ;
72AB  FD23        INC   IY             ;
72AD  FD          DEFB  0FDH           ; Sonderbefehl 'LD  A,LSB IY'
72AE  7D          LD    A,L            ;
72AF  FE68        CP    68H            ; = 68H (IY = 7868H) ?
72B1  C29D72      JP    NZ,729DH       ; wenn nicht, dann nach 729DH
72B4  E1          POP   HL             ; Stack korrigieren
72B5  210000      LD    HL,0000H       ; naechste Schussposition = 0
72B8  C9          RET                  ; und zurueck
72B9  E1          POP   HL             ; Trefferposition zurueck in HL
72BA  7E          LD    A,(HL)         ; getroffenes Zeichen in Akku
72BB  E601        AND   01H            ; mit 01H undieren. Bei Ergebnis 0 ist
72BD  2801        JR    Z,$+03H        ; es das linkesZeichen eines Feindes
72BF  2B          DEC   HL             ; sonst HL decrementiren
72C0  FD7500      LD    (IY+00H),L     ; LSB der Position in Tabelle ablegen
72C3  FD7401      LD    (IY+01H),H     ; MSB ebenso.
72C6  FD360200    LD    (IY+02H),00H   ; Explosionszaehler auf 0 setzen
72CA  EB          EX    DE,HL          ; Position in DE zwischenspeichern
72CB  FD218078    LD    IY,7880H       ; IY auf Postionstabelle der Feinde
72CF  FD6E00      LD    L,(IY+00H)     ; LSB der Position in L
72D2  FD6601      LD    H,(IY+01H)     ; MSB in H
72D5  DF          RST   18H            ; vergleich mit Trefferposition in DE
72D6  280E        JR    Z,$+10H        ; gleich ? dann nach 72E6H
72D8  010400      LD    BC,0004H       ; sonst IY +4
72DB  FD09        ADD   IY,BC          ; addieren
72DD  FD          DEFB  0FDH           ; LSB von IY in Akku
72DE  7D          LD    A,L            ;
72DF  B7          OR    A              ; Flags setzen
72E0  20ED        JR    NZ,$-11H       ; <> 0 ? dann nach 72CFH
```

```
72E2  210000    LD    HL,0000H      ; Naechste Schussposition gleich 0
72E5  C9        RET                 ; zurueck, kein Feind getroffen
72E6  3620      LD    (HL),20H      ; Feind im Bildschirm loeschen
72E8  23        INC   HL            ;
72E9  3620      LD    (HL),20H      ; zweites Byte loeschen
72EB  210000    LD    HL,0000H      ; 0 in Positionstabelle der Feinde
72EE  FD7500    LD    (IY+00H),L    ; schreiben. Der entsprechende Feind
72F1  FD7401    LD    (IY+01H),H    ; ist geloescht
72F4  2A0242    LD    HL,(4202H)    ; Punktzahl in HL
72F7  3A0042    LD    A,(4200H)     ; Level in Akku
72FA  07        RLCA                ; *2
72FB  3C        INC   A             ; +1
72FC  5F        LD    E,A           ; nach E speichern
72FD  1600      LD    D,00H         ; D loeschen
72FF  19        ADD   HL,DE         ; DE zur Punktzahl addieren
7300  220242    LD    (4202H),HL    ; und wieder abspeichern
7303  3A0B42    LD    A,(420BH)     ; Anzahl der verbliebenen Feinde in Akku
7306  3D        DEC   A             ; -1
7307  320B42    LD    (420BH),A     ; wieder abspeichern
730A  210000    LD    HL,0000H      ; naechste Schussposition gleich 0
730D  3E08      LD    A,08H         ; 08H        (SOUND 8,15 :
730F  D3F8      OUT   (0F8H),A      ; auf Port F8H   Lautstaerke von Kanal
7311  3E0F      LD    A,0FH         ; 0FH         1 auf maximum setzen)
7313  D3F9      OUT   (0F9H),A      ; auf Port F9H
7315  C9        RET                 ; und zurueck
```

; *** AUSGABE EINER ZAHL ***

```
7320  3E30      LD    A,30H         ; 30H = ASC("0") in Akku
7322  3C        INC   A             ; +1
7323  09        ADD   HL,BC         ; HL = HL + BC (BC = -10000,-1000 usw.)
7324  38FC      JR    C,$-02H       ; wenn Carry gesetzt dann weiter
7326  3D        DEC   A             ; sonst Akku -1
7327  ED42      SBC   HL,BC         ; HL = HL - BC
7329  12        LD    (DE),A        ; Akku auf Adresse (DE) abspeichern
732A  13        INC   DE            ; DE auf naechste Position
732B  C9        RET                 ; zurueck
732C  D5        PUSH  DE            ; Adresse fuer Ausgabe der Zahl retten
732D  ED5B0C42  LD    DE,(420CH)    ; Punktzahl fuer Bonusschiff nach DE
7331  DF        RST   18H           ; mit Punktzahl vergleichen
7332  381A      JR    C,$+1CH       ; Wenn groesser, dann nach 734EH
7334  EB        EX    DE,HL         ; Bonuspunktzahl nach HL
7335  01D007    LD    BC,07D0H      ; 07D0H = 2000D
7338  09        ADD   HL,BC         ; addieren
7339  220C42    LD    (420CH),HL    ; und abspeichern
733C  EB        EX    DE,HL         ; Punktzahl zurueck nach HL
733D  11E479    LD    DE,79E4H      ; DE auf Position der Schiffanzeige
7340  1A        LD    A,(DE)        ; Zeichen in Akku
7341  FEB0      CP    0B0H          ; Zeichen fuer Schiff ?
7343  2006      JR    NZ,$+08H      ; wenn nicht, dann nach 734BH
7345  13        INC   DE            ; naechste Position
7346  7B        LD    A,E           ; LSB in Akku
7347  FEEE      CP    0EEH          ; = EEH ? (Hier beginnt 'HIGH-SCORE..')
7349  20F5      JR    NZ,$-09H      ; wenn nicht, dann nach 7340H
734B  3EB0      LD    A,0B0H        ; Zeichen fuer Schiff in Akku
734D  12        LD    (DE),A        ; auf DE abspeichern
```

```
734E  D1        POP   DE           ; Adresse fuer Ausgabe zurueck
734F  01F0D8    LD    BC,0D8F0H    ; BC = -10000
7352  CD2073    CALL  7320H        ; Stelle berechnen
7355  0118FC    LD    BC,0FC18H    ; BC = -1000
7358  CD2073    CALL  7320H        ; Stelle berechnen
735B  019CFF    LD    BC,0FF9CH    ; BC = -100
735E  CD2073    CALL  7320H        ; Stelle berechnen
7361  01F6FF    LD    BC,0FFF6H    ; BC = -10
7364  CD2073    CALL  7320H        ; Stelle berechnen
7367  01FFFF    LD    BC,0FFFFH    ; BC = -1
736A  CD2073    CALL  7320H        ; Stelle berechnen
736D  210879    LD    HL,7908H     ; Adresse der Ausgabezeile
7370  110044    LD    DE,4400H     ; erste Bildschirmzeile
7373  012800    LD    BC,0028H     ; 40 Zeichen
7376  EDB0      LDIR               ; kopieren
7378  C9        RET                ; und zurueck
```

; *** UNTERPROGRAMM SCHUESSE UND EXPLOSIONEN BEWEGEN ***

```
7380  DD217078  LD    IX,7870H     ; IX auf Tabelle der Schuesse
7384  DD6E00    LD    L,(IX+00H)   ; LSB der Position nach L
7387  DD6601    LD    H,(IX+01H)   ; MSB nach H
738A  7C        LD    A,H          ; MSB
738B  B5        OR    L            ; oderiert mit LSB
738C  2836      JR    Z,$+38H      ; = 0 ? dann nach 73C4H
738E  7E        LD    A,(HL)       ; Zeichen auf Schussposition
738F  FEA0      CP    0A0H         ; < A0H ?
7391  381A      JR    C,$+1CH      ; wenn ja, dann nach 73ADH
7393  3620      LD    (HL),20H     ; Schuss loeschen
7395  11D8FF    LD    DE,0FFD8H    ; -40D
7398  19        ADD   HL,DE        ; zur Position addieren
7399  112844    LD    DE,4428H     ; 2. Bilschirmzeile
739C  DF        RST   18H          ; mit Position vergleichen
739D  3005      JR    NC,$+07H     ; wenn groesser dann nach 73A4H
739F  210000    LD    HL,0000H     ; sonst naechste Schussposition gleich 0
73A2  1814      JR    $+16H        ; und nach 73B8H
73A4  7E        LD    A,(HL)       ; Zeichen auf neuer Position in Akku
73A5  FE80      CP    80H          ; < 80H ? (min. der Zeichen fuer Feinde)
73A7  3807      JR    C,$+09H      ; wenn ja, dann nach 73B0H
73A9  FEA0      CP    0A0H         ; >= A0H ? (max.)
73AB  3003      JR    NC,$+05H     ; wenn nicht, dann nach 73B0H
73AD  CD9872    CALL  7298H        ; Unterprg.: Explosion initialisieren
73B0  3A0A42    LD    A,(420AH)    ; Zaehler in Akku
73B3  E601      AND   01H          ; mit 01H undieren (ergibt 0 oder 1)
73B5  C6A6      ADD   A,0A6H       ; + A6H (ergibt Zeichen fuer Schuss)
73B7  77        LD    (HL),A       ; auf neuer Position abspeichern
73B8  DD7500    LD    (IX+00H),L   ; LSB Schussposition wieder in Tabelle
73BB  DD7401    LD    (IX+01H),H   ; MSB ebenso
73BE  1100AC    LD    DE,0AC00H    ; Offset zum Farbspeicher
73C1  19        ADD   HL,DE        ; addieren
73C2  3609      LD    (HL),09H     ; und mit 09H belegen
73C4  DD23      INC   IX           ; Tabellenzeiger +2
73C6  DD23      INC   IX           ;
73C8  DD        DEFB  0DDH         ; LSB von IX in Akku
73C9  7D        LD    A,L          ;
73CA  FE80      CP    80H          ; = 80H (IX = 7880H)
```

```
73CC  C28473     JP    NZ,7384H      ; wenn nicht, nach 7384H
73CF  DD215078   LD    IX,7850H      ; IX auf Tabelle der Explosionen
73D3  DD6E00     LD    L,(IX+00H)    ; LSB der Position in L
73D6  DD6601     LD    H,(IX+01H)    ; MSB in H
73D9  7C         LD    A,H           ; MSB
73DA  B5         OR    L             ; oderiert mit LSB
73DB  2828       JR    Z,$+2AH       ; = 0 ? dann nach 7405H
73DD  DD7E02     LD    A,(IX+02H)    ; Explosionszaehler in Akku
73E0  07         RLCA                ; *2
73E1  C6AA       ADD   A,0AAH        ; + AAH (Zeichen der Explosionen)
73E3  FEB0       CP    0B0H          ; = B0H ? (Explosion zuende)
73E5  2008       JR    NZ,$+0AH      ; wenn nicht, nach 73EFH
73E7  3620       LD    (HL),20H      ; Explosion loeschen
73E9  23         INC   HL            ;
73EA  3620       LD    (HL),20H      ; zweites Byte loeschen
73EC  210000     LD    HL,0000H      ; Position gleich 0
73EF  DD7500     LD    (IX+00H),L    ; wieder abspeichern
73F2  DD7401     LD    (IX+01H),H    ;
73F5  77         LD    (HL),A        ; Zeichen auf Explosion abspeichern
73F6  23         INC   HL            ; naechste Adresse
73F7  3C         INC   A             ; naechstes Zeichen
73F8  77         LD    (HL),A        ; abspeichern
73F9  11FFAB     LD    DE,0ABFFH     ; Offset zum Farbspeicher -1 (wegen INC)
73FC  19         ADD   HL,DE         ; zur Position addieren
73FD  D609       SUB   09H           ; Zeichenwert -9 ergibt Farbe (2,4od. 6)
73FF  77         LD    (HL),A        ; in den Farbspeicher ablegen
7400  23         INC   HL            ; und
7401  77         LD    (HL),A        ; auf die naechste Adresse
7402  DD3402     INC   (IX+02H)      ; Explosionszaehler +1
7405  DD23       INC   IX            ; Tabellenzaehler +3
7407  DD23       INC   IX            ;
7409  DD23       INC   IX            ;
740B  DD         DEFB  0DDH          ; LSB von IX in Akku
740C  7D         LD    A,L           ;
740D  FE68       CP    68H           ; = 68H ?  (IX = 7868H)
740F  C2D373     JP    NZ,73D3H      ; wenn nicht, naechste Explosion
7412  2A0242     LD    HL,(4202H)    ; Punktzahl
7415  11DE79     LD    DE,79DEH      ; auf 79DEH (Ausgabezeile)
7418  CD2C73     CALL  732CH         ; berechnen und ausgeben
741B  2A0242     LD    HL,(4202H)    ; Punktzahl
741E  ED5B0442   LD    DE,(4204H)    ; und bestes Ergebnis
7422  DF         RST   18H           ; vergleichen
7423  3809       JR    C,$+0BH       ; wenn Punktzahl kleiner,dann nach 742EH
7425  220442     LD    (4204H),HL    ; sonst als bestes Ergebnis abspeichern
7428  11FB79     LD    DE,79FBH      ; und auf 79FBH (Ausgabezeile)
742B  CD4F73     CALL  734FH         ; berechnen und ausgeben
742E  C9         RET                 ; zurueck
```

; *** UNTERPROGRAMM SCHIFF TESTEN UND LOESCHEN ***

```
7438  2A0642     LD    HL,(4206H)    ; Schiffsposition in HL
743B  0603       LD    B,03H         ; Zaehler fuer Anzahl der Bytes
743D  7E         LD    A,(HL)        ; Zeichen aus Bildschirm in Akku
743E  FEA0       CP    0A0H          ; < A0H ? (gehoert zu Feinden)
7440  3809       JR    C,$+0BH       ; wenn ja, dann nach 744BH
7442  3620       LD    (HL),20H      ; sonst loeschen
```

```
7444   23          INC    HL              ; naechstes Byte
7445   10F6        DJNZ   $-08H           ; alle getestet ? wenn nicht, nach 743DH
7447   2B          DEC    HL              ; HL wieder auf die Position des
7448   2B          DEC    HL              ; ersten Zeichens bringen
7449   2B          DEC    HL              ;
744A   C9          RET                    ; und zurueck
744B   F1          POP    AF              ; Stack korrigieren
744C   C33075      JP     7530H           ; Unterprg.: Explosion des Schiffes
```

; *** SCHIFF UND FARBEN SETZEN ***

```
7450   06A0        LD     B,0A0H          ; A0H ist erste Phase des Schiffes
7452   3A0A42      LD     A,(420AH)       ; Zaehler in Akku
7455   E601        AND    01H             ; undieren mit 01H (ergibt 0 oder 1)
7457   2802        JR     Z,$+04H         ; = 0 ? dann nach 745BH
7459   06A3        LD     B,0A3H          ; sonst A3H in B (zweite Phase)
745B   78          LD     A,B             ; B in Akku
745C   77          LD     (HL),A          ; auf Schiffsposition
745D   23          INC    HL              ; naechste Adresse
745E   3C          INC    A               ; naechstes Zeichen
745F   77          LD     (HL),A          ; abspeichern
7460   23          INC    HL              ; naechste Adresse
7461   3C          INC    A               ; naechstes Zeichen
7462   77          LD     (HL),A          ; abspeichern
7463   11FEAB      LD     DE,0ABFEH       ; Offset zum Farbspeicher -2
7466   19          ADD    HL,DE           ; zur Position addieren
7467   3606        LD     (HL),06H        ; 06H abspeichern (orange)
7469   23          INC    HL              ; naechste Adresse
746A   3602        LD     (HL),02H        ; 02H abspeichern (rot)
746C   23          INC    HL              ; naechste Adresse
746D   3606        LD     (HL),06H        ; 06H abspeichern (orange)
746F   C9          RET                    ; und Zurueck
```

; *** BILDAUFBAU UND SPIELSTART ***

```
7480   210044      LD     HL,4400H        ; Erste Bildschirmposition
7483   110144      LD     DE,4401H        ; Zweite Bildschirmposition
7486   01FF03      LD     BC,03FFH        ; Bildschirmlaenge -1
7489   3620        LD     (HL),20H        ; 20H auf erstes Zeichen (SPACE)
748B   EDB0        LDIR                   ; 1. Zeichen auf 2. schieben usw.
748D   21D879      LD     HL,79D8H        ; Ausgabezeile
7490   110044      LD     DE,4400H        ; auf erste Bildschirmzeile
7493   012800      LD     BC,0028H        ; 40D Zeichen
7496   EDB0        LDIR                   ; schieben
7498   210078      LD     HL,7800H        ; Farben der ersten Zeile
749B   1100F0      LD     DE,0F000H       ; in den Farbspeicher
749E   012800      LD     BC,0028H        ; 40D Bytes
74A1   EDB0        LDIR                   ; schieben
74A3   3A0142      LD     A,(4201H)       ; Feindtyp in Akku
74A6   3C          INC    A               ; +1
```

```
74A7  E607      AND   07H          ; mit 07H undieren (nur 0..7 zugel.)
74A9  320142    LD    (4201H),A    ; und wieder abspeichern
74AC  21AB47    LD    HL,47ABH     ; Schiffsposition, vorletzte Zeile,Mitte
74AF  220642    LD    (4206H),HL   ; abspeichern
74B2  210079    LD    HL,7900H     ; Tabelle der Feindpositionen
74B5  118078    LD    DE,7880H     ; in die Arbeitstabelle
74B8  018000    LD    BC,0080H     ; 128D Bytes (fuer 32D Feinde)
74BB  EDB0      LDIR               ; schieben
74BD  210100    LD    HL,0001H     ; X-Richtung (0001H-rechts, FFFFH-links)
74C0  220842    LD    (4208H),HL   ; abspeichern
74C3  215078    LD    HL,7850H     ; Schuesse und Explosionen
74C6  115178    LD    DE,7851H     ; mit 00H ueberschreiben
74C9  012F00    LD    BC,002FH     ; (0000H als Position bedeutet Schuss,
74CC  3600      LD    (HL),00H     ; Explosion oder Feind existiert nicht)
74CE  EDB0      LDIR               ;
74D0  3E20      LD    A,20H        ; Anzahl der Feinde (32D) abspeichern
74D2  320B42    LD    (420BH),A    ;
74D5  3A0042    LD    A,(4200H)    ; Level in Akku
74D8  3C        INC   A            ; +1
74D9  320042    LD    (4200H),A    ; und wieder abspeichern
74DC  212878    LD    HL,7828H     ; Text '** GET READY **'
74DF  113C46    LD    DE,463CH     ; in Bildschirmspeicher
74E2  011000    LD    BC,0010H     ; 16 Zeichen
74E5  EDB0      LDIR               ; schieben
74E7  2A0642    LD    HL,(4206H)   ; Schiffsposition in HL
74EA  CD5074    CALL  7450H        ; Schiff und Farben setzen
74ED  3E18      LD    A,18H        ; 18H = JR unbedingt, damit die Feinde
74EF  328971    LD    (7189H),A    ; im Vorspann keine Y-Richtung bekommen
74F2  060C      LD    B,0CH        ; Zaehler (fuer 12D Durchlaeufe)
74F4  C5        PUSH  BC           ; retten

74F5  CD0071    CALL  7100H        ; Unterprg.: Feinde bewegen
74F8  CDC071    CALL  71C0H        ; Unterprg.: X-Richtung testen
74FB  213CF2    LD    HL,0F23CH    ; Farbspeicheradresse von 'GET READY'
74FE  113DF2    LD    DE,0F23DH    ; mit einen
7501  010F00    LD    BC,000FH     ; zufaelligen Wert
7504  ED5F      LD    A,R          ; aus dem R-(Refresh-) Register
7506  77        LD    (HL),A       ; belegen
7507  EDB0      LDIR               ; 16D Bytes
7509  010018    LD    BC,1800H     ; Warteschleifenwert
750C  CD6000    CALL  0060H        ; ROM-Routine: Schleife auf BC
750F  C1        POP   BC           ; BC zurueck vom Stack
7510  10E2      DJNZ  $-1CH        ; und evtl. neuer Durchlauf
7512  3E20      LD    A,20H        ; 20H = JR NZ (non zero), die Feinde
7514  328971    LD    (7189H),A    ; koennen sich jetzt wieder voll bewegen
7517  213C46    LD    HL,463CH     ; '** GET READY **' Position im Bild-
751A  113D46    LD    DE,463DH     ; schirmspeicher mit 20H (SPACE)
751D  010F00    LD    BC,000FH     ; belegen
7520  77        LD    (HL),A       ; im Akku steht noch 20H
7521  EDB0      LDIR               ; 16D Bytes
7523  C30072    JP    7200H        ; Sprung zur Hauptschleife
```

```
; *** EXPLOSION DES SCHIFFES ***
```

```
7530  061F      LD    B,1FH        ; Zaehler (31 Durchlaeufe)
```

```
7532  3E07     LD    A,07H        ; 07H         (SOUND 7,23)
7534  D3F8     OUT   (0F8H),A     ; auf Port F8H    Kanal 1 und 2 werden
7536  3EE7     LD    A,0E7H       ; E7H         auf Rauschgenerator
7538  D3F9     OUT   (0F9H),A     ; auf Port F9H    geschaltet)
753A  C5       PUSH  BC           ; Zaehler retten
753B  3E09     LD    A,09H        ; 09H         (SOUND 9,0..15
753D  D3F8     OUT   (0F8H),A     ; auf Port F8H    Lautstaerke von
753F  78       LD    A,B          ; Zaehler in Akku  Kanal 2 wird mit
7540  E60F     AND   0FH          ; mit 0FH undieren  dem Schleifenzaehler
7542  D3F9     OUT   (0F9H),A     ; auf Port F9H    geaendert)
7544  3E06     LD    A,06H        ; 06H
7546  D3F8     OUT   (0F8H),A     ; auf Port F8H    (SOUND 6,0..31
7548  3E1F     LD    A,1FH        ; 1FH in Akku     Rauschfrequenz wird
754A  90       SUB   B            ; minus Zaehler    mit dem Zaehler ge-
754B  D3F9     OUT   (0F9H),A     ; auf Port F9H    aendert)
754D  3A0A42   LD    A,(420AH)    ; Zaehler in Akku
7550  3C       INC   A            ; +1
7551  320A42   LD    (420AH),A    ; und wieder abspeichern
7554  CD0071   CALL  7100H        ; Unterprg.: Feinde bewegen
7557  CDC071   CALL  71C0H        ; Unterprg.: X-Richtung testen
755A  CD8073   CALL  7380H        ; Unterprg.: Schuesse und Explosionen
755D  2A5078   LD    HL,(7850H)   ; Position der ersten Explosion in HL
7560  7C       LD    A,H          ; MSB
7561  B5       OR    L            ; oderiert mit LSB
7562  2018     JR    NZ,$+1AH     ; <> 0 ? dann nach 757CH
7564  2A0642   LD    HL,(4206H)   ; Schiffsposition in HL
7567  ED5F     LD    A,R          ; R-Register als Zufallswert
7569  E601     AND   01H          ; mit 01H undieren
756B  2801     JR    Z,$+03H      ; = 0 ? dann nach 756EH
756D  23       INC   HL           ; sonst Position +1 (rechter Teil)
756E  225078   LD    (7850H),HL   ; in Explosionstabelle abspeichern
7571  325278   LD    (7852H),A    ; Akku in Explosionszaehler
7574  3E08     LD    A,08H        ; 08H             (SOUND 8,15
7576  D3F8     OUT   (0F8H),A     ; auf Port F8H    Kanal 1 auf maximale
7578  3E0F     LD    A,0FH        ; 0FH         Lautstaerke, Explo-
757A  D3F9     OUT   (0F9H),A     ; auf Port F9H    sionsgeraeusch)
757C  C1       POP   BC           ; Schleifenzaehler zurueck
757D  10BB     DJNZ  $-43H        ; fertig ? wenn nicht, dann nach 753AH
757F  21EE79   LD    HL,79EEH     ; Letztes Schiff in der Ausgabezeile
7582  7E       LD    A,(HL)       ; Zeichen in Akku

7583  FEB0     CP    0B0H         ; = B0H (Zeichen fuer Schiff)
7585  2809     JR    Z,$+0BH      ; wenn ja, dann nach 7590H
7587  2B       DEC   HL           ; ein Zeichen vor
7588  7D       LD    A,L          ; LSB in Akku
7589  FEE3     CP    0E3H         ; = Position des ersten Schiffes -1 ?
758B  20F5     JR    NZ,$-09H     ; wenn nicht, dann nach 7582H
758D  C3A075   JP    75A0H        ; sonst Spielende
7590  3620     LD    (HL),20H     ; Schiff in Ausgabezeile loeschen
7592  3A0142   LD    A,(4201H)    ; Feindtyp in Akku
7595  3D       DEC   A            ; -1
7596  320142   LD    (4201H),A    ; wieder abspeichern
7599  C38074   JP    7480H        ; und das Bild wieder aufbauen
```

```
; *** SPIELENDE ***
```

```
75A0  21FA79    LD    HL,79FAH       ; bestes Ergebnis von der Ausgabezeile
75A3  11607D    LD    DE,7D60H       , in die Anfangsgrafik kopieren
75A6  010600    LD    BC,0006H       ,
75A9  EDB0      LDIR                 ,
75AB  C30070    JP    7000H          , Zur Eingabeschleife und neues Spiel


; *** PROGRAMMSTART ***


75B0  210000    LD    HL,0000H       , High-Score
75B3  220442    LD    (4204H),HL     , loeschen
75B6  310060    LD    SP,6000H       , Stackpointer auf 6000H (vor Programm)
75B9  C30070    JP    7000H          , und Sprung zur Eingabeschleife
```

** DER ZEICHENSATZ DES PROGRAMMS ZALAGA **

CHR$(128) CHR$(129) CHR$(130) CHR$(131) CHR$(132) CHR$(133)

CHR$(134) CHR$(135) CHR$(136) CHR$(137) CHR$(138) CHR$(139)

CHR$(140) CHR$(141) CHR$(142) CHR$(143) CHR$(144) CHR$(145)

CHR$(146) CHR$(147) CHR$(148) CHR$(149) CHR$(150) CHR$(151)

CHR$(152) CHR$(153) CHR$(154) CHR$(155) CHR$(156) CHR$(157)

CHR$(158) CHR$(159) CHR$(160) CHR$(161) CHR$(162) CHR$(163)

CHR$(164)	CHR$(165)	CHR$(166)	CHR$(167)	CHR$(168)	CHR$(169)
CHR$(170)	CHR$(171)	CHR$(172)	CHR$(173)	CHR$(174)	CHR$(175)
CHR$(176)	CHR$(177)	CHR$(178)	CHR$(179)	CHR$(180)	CHR$(181)
CHR$(182)	CHR$(183)	CHR$(184)	CHR$(185)	CHR$(186)	CHR$(187)
CHR$(188)	CHR$(189)	CHR$(190)	CHR$(191)	CHR$(192)	CHR$(193)

Ein schnelles Formatierprogramm für das Colour-Genie
(nur für Diskettenbenutzer)

Sicher ist Ihnen nicht entgangen, daß die Formatierroutine des Colour-DOS einen sehr großen Nachteil hat: sie ist zu langsam! Wir präsentieren hier ein Programm, das Disketten auf dem Colour-Genie in einem Bruchteil der gewohnten Zeit formatiert. Dieses Programm besteht aus zwei Teilen:

1. Ein Maschinenspracheprogramm. Geben Sie dieses bitte auf irgendeine Art (z.B. mit dem Basic-Monitor, den Sie weiter vorne in diesem Buch gelistet finden: für diesen sind auch die Prüfsummen am Ende jeder Zeile bestimmt, diese also nicht (!) mit eingeben.) in den Rechner ein. Dann speichern Sie dieses Programm mit
   CMD"W FFORMAT/CMD,&H8000,&H82A7,&H0066"
   auf Diskette ab. Geben Sie nun das kleine BASIC-Programm ein, das hinter dem Hex-Listing steht. Dieses Programm lädt und startet dann das Formatier-Programm.

```
8000 2B 3E 2C 20 3C 3B 3E 2C   96
8008 20 3C 40 3E 2C 20 3C 53   B5
8010 70 61 28 4E 16 4E 0C 00   B7
8018 03 F5 00 01 16 4E 0C 00   69
8020 03 F5 00 00 14 FF 0B FF   15
8028 06 00 00 01 0B FF 06 00   17
8030 00 00 D5 C5 3A 0E 5A CB   07
8038 3F 47 57 21 00 80 7A 90   88
8040 77 23 82 77 23 10 F7 21   DE
8048 00 80 D9 11 A8 82 21 14   C9
8050 80 3A 11 5A E6 01 20 03   2F
8058 21 26 80 2B 7E 32 AE 80   D0
8060 2B CD B8 80 3A 0E 5A 4F   21
8068 E5 E1 E5 CD B8 80 20 FB   CB
8070 EB 36 FE 23 3A A7 82 77   1C
8078 23 36 00 23 D9 7E 23 D9   CF
8080 77 23 36 01 23 36 F7 23   44
8088 EB CD B8 80 20 FB EB 36   2C
8090 FB 23 06 00 36 E5 23 10   72
8098 FB 36 F7 23 EB 0D C2 69   6E
80A0 80 21 A8 A2 AF ED 52 23   FC
80A8 44 4D D5 E1 13 36 4E ED   CB
80B0 B0 E1 21 A8 82 C1 D1 C9   37
80B8 C5 46 23 4E 23 78 B7 28   F6
80C0 07 79 12 13 10 FC AF 3D   9D
80C8 C1 C9 E5 D5 C5 CD BB D0   61
80D0 11 EF FF 21 EC FF 01 A8   B4
80D8 82 36 F0 CD 37 D1 0A CB   52
80E0 46 28 27 CB 4E 28 F8 12   E0
```

```
80E8 03 0A CB 4E CA EA 80 12   6C
80F0 03 0A CB 4E 20 F9 CB 4E   58
80F8 20 F5 CB 4E 20 F1 CB 46   50
8100 28 08 CB 4E 20 E9 CB 7E   9B
8108 28 E8 7E 36 D0 E6 F8 C1   33
8110 D1 E1 C9 3A A7 82 21 E3   E2
8118 81 CD 05 82 21 D3 81 ED   37
8120 5B 20 40 01 0A 00 ED B0   63
8128 21 DD 81 01 13 00 ED B0   30
8130 CD 32 80 06 03 CD CA 80   9F
8138 28 14 10 F9 47 3E 10 CB   A5
8140 78 C2 14 82 3D CB 70 C2   0A
8148 14 82 3D C3 14 82 21 F0   3D
8150 81 ED 5B 20 40 01 0A 00   34
8158 ED B0 3A 0E 5A 47 11 00   97
8160 80 C5 1A 21 03 82 CD 05   D7
8168 82 2A 20 40 D5 11 12 00   04
8170 19 EB 21 FA 81 01 0B 00   AC
8178 ED B0 D1 06 03 21 A8 82   C2
8180 1A CD 5B 82 E6 1C 28 06   F4
8188 10 F3 C1 C3 13 81 13 C1   EF
8190 10 CF C9 CD 99 D0 AF 32   BF
8198 A7 82 32 ED FF 3A 0D 5A   EB
81A0 47 C5 3A 0A 5A E6 40 28   F8
81A8 08 21 09 5A CB E6 CD 13   1D
81B0 81 21 09 5A CB A6 CD 13   56
81B8 81 0E 43 CD 9B D0 3A A7   EB
81C0 82 3C 32 A7 82 32 ED FF   37
81C8 C1 10 D6 3E 0D CD 33 00   F2
81D0 C3 99 D0 46 6F 72 6D 61   21
81D8 74 69 6E 67 20 54 72 61   F9
81E0 63 6B 20 30 30 20 20 20   AE
81E8 20 20 20 20 20 20 20 20   00
81F0 56 65 72 69 66 79 69 6E   4C
81F8 67 20 2C 20 53 65 63 74   62
8200 6F 72 20 30 30 C5 06 2F   5B
8208 04 D6 0A 30 FB C6 3A 70   7F
8210 23 77 C1 C9 6F 26 00 CD   86
8218 9A 0A ED 7B 20 82 E1 C9   58
8220 00 00 E5 ED 73 20 82 CD   B4
8228 7F 0A 7D CD 40 D1 C2 14   BA
8230 82 F3 3E C9 32 12 40 AF   AF
8238 32 23 40 ED 7B 81 40 CD   BB
8240 C9 01 CD 93 81 AF 32 ED   79
8248 FF 11 00 00 CD 03 DE 21   DF
8250 00 00 CD 9A 0A ED 7B 20   F9
8258 82 E1 C9 E5 D5 C5 E5 C1   51
8260 F5 11 EF FF 21 EC FF CD   CD
8268 BB D0 F1 32 EE FF 3A A7   7C
8270 82 32 ED FF 36 88 CD 37   62
8278 D1 CB 46 28 21 3E 83 A6   92
```

```
8280 E2 7D 82 1A 02 03 CB 4E   19
8288 20 F9 CB 4E 20 F5 CB 4E   60
8290 20 F1 CB 46 28 08 CB 4E   6B
8298 20 E9 CB 7E 28 E8 7E 36   16
82A0 D0 E6 3C C1 D1 E1 C9 00   2E
82A8
```

Und hier das dazugehoerige BASIC-Programm:

```
10 CLS:PRINT"Fast Format":PRINT:PRINT
20 INPUT"Welches Laufwerk";A
30 CMD"L FFORMAT/CMD"
40 DEFUSR1=&H8222
50 X=USR1(A)
60 IFX<>0THENPRINT"Disk error No.";X
70 END
```

Erlauterung des Maschinenprogramms "FAST FORMAT"

Das Programm wird mit dem USR-Befehl aufgerufen. Es wird zur Routine MAIN gesprungen. Dort wird das HL-Register (der Zeiger auf den Programmtext) gerettet. Die Routine 0A7FH liest das Argument des USR-Befehls als Integerzahl in das HL-Register. interessant ist davon aber nur das niederwertige Byte, die Laufwerksnummer, in der die zu formatierende Diskette liegt. Die Routine MOUNT (eine DOS-Routine, die bei 0140H beginnt), selektiert dieses Laufwerk und uberpruft, ob eine Diskette in diesem Laufwerk liegt und die Laufwerksklappe geschlossen ist. Sollte dies nicht der Fall sein, wird zur Fehlerbehandlungsroutine gesprungen. Sonst werden die Interrupts abgeschaltet, der Bildschirm geloscht (01C9H) und die eigentliche Formatroutine aufgerufen. Zum Schluß richtet das DOS das Inhaltsverzeichnis der Diskette ein (CALL 0DE03H). Wichtig: Beim Sprung auf diese Routine mussen das A, E und das D-Register den Wert Null enthalten. Schließlich, wenn alles gutgegangen ist, wird die Fehlernummer Null (kein Fehler) durch CALL 0A9AH zuruckgegeben. Nun die Erlauterung der einzelnen Unterprogramme:

BUILD: Dieses Programm baut ab der Adresse BUFFER den zu schreibenden Track auf. Das Aussehen des Tracks ist variabel, es hangt von der Einstellung fur dieses Laufwerk ab:

SINGLE DENSITY:

	20 Bytes	0FFH
pro Sektor:	11 Bytes	0FFH
	6 Bytes	00H
	1 Byte	0FEH
	1 Byte	Spurnummer
	1 Byte	Seitennummer (immer Null)
	1 Byte	Sektornummer
	1 Byte	Sektorlange (siehe unten)
	1 Byte	0F7H (generiert Prufsumme)
	11 Bytes	0FFH
	6 Bytes	00H
	1 Byte	0FBH
	n Bytes	0E5H (n ist Sektorlange)
	1 Byte	0F7H (generiert Prufsumme)

Diese Daten werden fur jeden Sektor geschrieben. Dann folgen Bytes 0FFH, bis die gesamte Spur 2000H Bytes lang ist.

DOUBLE DENSITY:

	40 Bytes	4EH
pro Sektor:	22 Bytes	4EH
	12 Bytes	00H
	3 Bytes	0F5H
	1 Byte	0FEH
	1 Byte	Spurnummer
	1 Byte	Seitennummer (immer Null)
	1 Byte	Sektornummer
	1 Byte	Sektorlänge (siehe unten)
	1 Byte	0F7H (generiert Prüfsumme)
	22 Bytes	4EH
	12 Bytes	00H
	3 Bytes	0F5H
	1 Byte	0FBH
	n Bytes	Daten (n = Sektorlänge)
	1 Byte	0F7H (generiert Prüfsumme)

Diese Daten werden für jeden Sektor geschrieben, dann folgen Bytes 4EH, bis die Spur 2000H Bytes im Speicher belegt.

Die Sektorlänge bedeutet folgendes:
00 = 128 Bytes
01 = 256 Bytes
02 = 512 Bytes
03 = 1024 Bytes

WRTTRK: Die Routine WRTTRK schreibt den vorher vorbereiteten Speicherbereich ab BUFFER auf die Spur, auf der momentan der Schreib-/Lesekopf des Laufwerks steht.

FMSUB: Dieses Programm baut eine Spur im Speicher auf, schreibt sie auf die Diskette und versucht dann, jeden einzelnen Sektor wieder einzulesen, um sicherzustellen, daß die Formatierung einwandfrei verlaufen ist.

FORMAT: FORMAT formatiert eine ganze Diskette. Die Routine FMSUB wird für jede einzelne Spur aufgerufen, dann wird der Schreib-/Lesekopf des Laufwerks eine Spur weiter nach innen gesetzt. Dies wird sooft wiederholt, wie die Diskette Spuren hat.

RDSEC: RDSEC liest einen Sektor von der Spur, auf der der Schreib-/Lesekopf momentan steht.

```
        org   8000h
sectab: defs 18
        defb 40,4eh
secdd:  defb 22,4eh,12,0,3,0f5h,0,1,22,4eh,12,0,3,0f5h,0,0
        defb 20,0ffh
secsd:  defb 11,0ffh,6,0,0,1,11,0ffh,6,0,0,0
build:  push de
        push bc
        ld    a,(5a0eh)
        srl   a
        ld    b,a
        ld    d,a
        ld    hl,sectab
by:     ld    a,d
        sub   b
        ld    (hl),a
        inc   hl
        add   a,d
        ld    (hl),a
        inc   hl
        djnz by
        ld    hl,sectab
        exx
        ld    de,buffer
        ld    hl,secdd
        ld    a,(5a11h)
        and   1
        jr    nz,bx
        ld    hl,secsd
bx:     dec   hl
        ld    a,(hl)
        ld    (filler),a
        dec   hl
        call wrtstr
        ld    a,(5a0eh)
        ld    c,a
        push hl
build1: pop   hl
        push hl
build2: call wrtstr
        jr    nz,build2
        ex    de,hl
        ld    (hl),0feh
        inc   hl
        ld    a,(trksec)
        ld    (hl),a
        inc   hl
        ld    (hl),0
        inc   hl
        exx
        ld    a,(hl)
        inc   hl
        exx
        ld    (hl),a
        inc   hl
```

```
        ld   (hl),1
        inc  hl
        ld   (hl),0f7h
        inc  hl
        ex   de,hl
build3: call wrtstr
        jr   nz,build3
        ex   de,hl
        ld   (hl),0fbh
        inc  hl
        ld   b,0
build4: ld   (hl),0e5h
        inc  hl
        djnz build4
        ld   (hl),0f7h
        inc  hl
        ex   de,hl
        dec  c
        jp   nz,buildf
        ld   hl,buffer+2000h
        xor  a
        sbc  hl,de
        inc  hl
        ld   b,h
        ld   c,l
        push de
        pop  hl
        inc  de
        ld   (hl),04eh
filler  equ  $-1
        ldir
        pop  hl
        ld   hl,buffer
        pop  bc
        pop  de
        ret
wrtstr: push bc
        ld   b,(hl)
        inc  hl
        ld   c,(hl)
        inc  hl
        ld   a,b
        or   a
        jr   z,wrtst2
        ld   a,c
wrtst1: ld   (de),a
        inc  de
        djnz wrtst1
        xor  a
        dec  a
wrtst2: pop  bc
        ret
wrttrk: push hl
        push de
        push bc
```

```
        call resel+5
        ld   de,0ffefh
        ld   hl,0ffech
        ld   bc,buffer
        ld   (hl),0f0h
        call wait
        ld   a,(bc)
wrt1:   bit  0,(hl)
        jr   z,wrt3
        bit  1,(hl)
        jr   z,wrt1
        ld   (de),a
        inc  bc
        ld   a,(bc)
wrt11:  bit  1,(hl)
        jp   z,wrt11
wrt12:  ld   (de),a
        inc  bc
        ld   a,(bc)
wrt13:  bit  1,(hl)
        jr   nz,wrt12
        bit  1,(hl)
        jr   nz,wrt12
        bit  1,(hl)
        jr   nz,wrt12
        bit  0,(hl)
        jr   z,wrt3
        bit  1,(hl)
        jr   nz,wrt12
        bit  7,(hl)
        jr   z,wrt13
wrt3:   ld   a,(hl)
        ld   (hl),0d0h
        and  0f8h
        pop  bc
        pop  de
        pop  hl
        ret
fmsub:  ld   a,(trksec)
        ld   hl,str3
        call convrt
        ld   hl,str1
        ld   de,(4020h)
        ld   bc,10
        ldir
        ld   hl,str2
        ld   bc,19
        ldir
        call build
        ld   b,3
form2:  call wrttrk
        jr   z,form3
        djnz form2
        ld   b,a
        ld   a,16
```

```
        bit  7,b
        jp   nz,dskerr
        dec  a
        bit  6,b
        jp   nz,dskerr
        dec  a
        jp   dskerr
form3:  ld   hl,str4
        ld   de,(4020h)
        ld   bc,10
        ldir
        ld   a,(5a0eh)
        ld   b,a
        ld   de,sectab
form4:  push bc
        ld   a,(de)
        ld   hl,str5+9
        call convrt
        ld   hl,(4020h)
        push de
        ld   de,18
        add  hl,de
        ex   de,hl
        ld   hl,str5
        ld   bc,11
        ldir
        pop  de
        ld   b,3
form5:  ld   hl,buffer
        ld   a,(de)
        call rdsec
        and  1ch
        jr   z,form6
        djnz form5
        pop  bc
        jp   fmsub
form6:  inc  de
        pop  bc
        djnz form4
        ret
format: call restor
        xor  a
        ld   (trksec),a
        ld   (0ffedh),a
        ld   a,(5a0dh)
        ld   b,a
form1:  push bc
        ld   a,(5a0ah)
        and  40h
        jr   z,form11
        ld   hl,5a09h
        set  4,(hl)
        call fmsub
form11: ld   hl,5a09h
        res  4,(hl)
```

```
        call fmsub
        ld   c,43h
        call isscmd
        ld   a,(trksec)
        inc  a
        ld   (trksec),a
        ld   (0ffedh),a
        pop  bc
        djnz form1
        ld   a,13
        call 33h
        jp   restor
str1:   defm 'Formating '
str2:   defm 'Track '
str3:   defm '00
str4:   defm 'Verifying '
str5:   defm ', Sector 00'
convrt: push bc
        ld   b,2fh
conv1:  inc  b
        sub  10
        jr   nc,conv1
        add  a,3ah
        ld   (hl),b
        inc  hl
        ld   (hl),a
        pop  bc
        ret
dskerr: ld   l,a
        ld   h,0
        call 0a9ah
        ld   sp,(stack)
        pop  hl
        ret
stack:  defw 0
main:   push hl
        ld   (stack),sp
        call 0a7fh
        ld   a,l
        call mount
        jp   nz,dskerr
        di
        ld   a,0c9h
        ld   (4012h),a
        xor  a
        ld   (4023h),a
        ld   sp,(40b1h)
        call 01c9h
        call format
        xor  a
        ld   (0ffedh),a
        ld   de,0
        call 0de03h
        ld   hl,0
        call 0a9ah
```

```
        ld   sp,(stack)
        pop  hl
        ret
rdsec:  push hl
        push de
        push bc
        push hl
        pop  bc
        push af
        ld   de,0ffefh
        ld   hl,0ffech
        call resel+5
        pop  af
        ld   (0ffeeh),a
        ld   a,(trksec)
        ld   (0ffedh),a
        ld   (hl),88h
        call wait
        bit  0,(hl)
        jr   z,rdsc3
rdsc0:  ld   a,83h
        and  (hl)
        jp   po,rdsc0
rdsc1:  ld   a,(de)
        ld   (bc),a
        inc  bc
rdsc2:  bit  1,(hl)
        jr   nz,rdsc1
        bit  1,(hl)
        jr   nz,rdsc1
        bit  1,(hl)
        jr   nz,rdsc1
        bit  0,(hl)
        jr   z,rdsc3
        bit  1,(hl)
        jr   nz,rdsc1
        bit  7,(hl)
        jr   z,rdsc2
rdsc3:  ld   a,(hl)
        ld   (hl),0d0h
        and  3ch
        pop  bc
        pop  de
        pop  hl
        ret
trksec: defb 0
select  equ  0d0c7h
resel   equ  0d0b6h
wait    equ  0d137h
isscmd  equ  0d09bh
restor  equ  0d099h
mount   equ  0d140h
buffer  equ  $
        end  main
```

Sektoren lesen und schreiben - vom Basic aus

Dieses kleine Programm konnen Sie in Ihre eigenen BASIC-Programme einbauen. Lassen Sie dann die Zeilen 30 bis 60 weg und ubergeben Sie die in der REM-Zeile genannten Argumente an das Programm. Zum Programm noch folgendes:

Das eigentliche Programm ist ein kurzes Maschinenspracheprogramm, das in A$ eingelesen wird. Es sieht folgendermaßen aus:

```
LD   HL,adresse
LD   DE,disk-relativer-sektor
CALL unterprogramm
LD   L,A
LD   H,0
JP   0A9AH
```

Die Werte fur adresse, disk-relativer-sektor und unterprogramm werden dann entsprechend Ihren Eingaben gePOKEd. Dieses Programm liest bzw. schreibt dann einen Sektor und gibt als Ergebnis des USR-Aufrufs die Nummer des aufgetretenen Fehlers an (0 = kein Fehler).

Programmlisting:

```
10 REM **** AD=address, SN=sector #, FL: 1=read, 2=write, 3=wp, DR=Lau
fwerksnummer
20 CLS:PRINT"Sektor In-/Output":PRINT:PRINT
30 INPUT"Laufwerk (0-3)";DV:IFDV<0ORDV>3THEN30
40 INPUT"Adresse";AD
50 INPUT"Diskrelativer Sektor";SN
60 INPUT"(1) Lesen, (2) Schreiben, (3) Lese-      geschuetzt schreiben"
:FL
70 RESTORE
80 READA$:IFA$<>"SECIO"THEN80
90 A$="":FORX=1TO15:READA:A$=A$+CHR$(A):NEXTX:FORX=1TOFL:READDR:NEXTX
100 X=VARPTR(A$):X=PEEK(X+1)+256*PEEK(X+2):IFX>32767THENX=X-65536
110 POKEX+1,(AD AND 255):POKE X+2,(INT(AD/256)AND255)
120 POKEX+4,(SN AND 255):POKE X+5,(INT(SN/256)AND255)
130 POKEX+7,(DR AND 255):POKE X+8,(INT(DR/256)AND255)
140 POKE&H5A08,DV:DEFUSR1=X:X=USR1(0):IFX=0THENEND
150 PRINT"Disk Error No.";X:END
160 DATASECIO
170 DATA33,0,0,17,0,0,205,0,0,111,38,0,195,154,10
180 DATA-12433,-12417,-12421
```

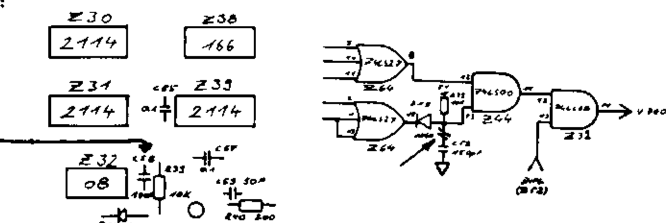
Zwei Tips fur den Betrieb eines einfarbigen Monitors am Colour-Genie

Viele Colour-Genie-Benutzer habe an Ihren Computer einen einfarbigen Monitor angeschlossen, da diese billiger als ein RGB-Farbmonitor sind, aber ein besseres Bild liefern, als ein Fernseher.
In diesem Falle kann man die Schrift noch verbessern, indem man

1.) Alle Farben auf hellweiß umsetzt (COLOUR 16).
Dies geht bei Basic-Programmen sehr einfach:
In den Adressen 4390H bis 439FH legt das Basic eine Farb-Tabelle an, die festlegt, welche Farbe durch welchen Colour-Befehl gewahlt wird. Alles was Sie nun machen mussen, ist in diese Tabelle uberall den 15 POKEn (Dies entspricht Hellweiß, s. auch Anhang C).
Also:
FORA=&H4390 TO &H439F : POKE A, 15 : NEXT
Bei Maschinenspracheprogrammen und Programmen, die direkt in den Farbspeicher POKEn, hilft dies i.d.R. nicht.

2.) Eine kleinen Gerateumbau vornimmt:
In der Video-Schaltung des Colour-Genies werden die Matrix-Punkte aller Zeichen verbreitert, damit bei Betrieb mit einem Farb-Fernseher die Punkte groß genug sind, um von der Lochmaskenrohre aufgelost werden zu konnen.
Dies ist bei einem Monitor naturlich uberflussig.
Diese Punktverbreiterung kann durch Entfernen eines Kondensators ruckgangig gemacht werden.
Dazu gehen Sie folgendermaßen vor:

Losen Sie die 3 Schrauben vorne unten und offnen Sie das Colour-Genie - Achten Sie darauf, daß Sie das Kabel des Level-Meters (wenn eingebaut) und das der Tastatur nicht abziehen.
Sie sehen vorne, in der Mitte der Hauptplatine, diese Bauteilgruppe:



Schneiden Sie nun mit einem feinen Seitenschneider eines der Anschlußbeine des Kondensators C58 durch und biegen Sie den Kondensators etwas hoch. (So konnen Sie den Kondensator jederzeit wieder anloten.)
Schrauben Sie das Gerat wieder zu. Fertig !
(Etwas Elektronik-Erfahrung sollten Sie fur diesen Umbau haben - sonst lassen Sie einen kundigen Bekannten die Umrustung vornehmen.)

Anhang A Colour-Basic-Tokens

Basic Einfachtokens

128	END	129	FOR
130	RESET	131	SET
132	CLS	133	CMD
134	RANDOM	135	NEXT
136	DATA	137	INPUT
138	DIM	139	READ
140	LET	141	GOTO
142	RUN	143	IF
144	RESTORE	145	GOSUB
146	RETURN	147	REM
148	STOP	149	ELSE
150	TRON	151	TROFF
152	DEFSTR	153	DEFINT
154	DEFSNG	155	DEFDBL
156	LINE	157	EDIT
158	ERROR	159	RESUME
160	OUT	161	ON
162	OPEN	163	FIELD
164	GET	165	PUT
166	CLOSE	167	LOAD
168	MERGE	169	NAME
170	KILL	171	LSET
172	RSET	173	SAVE
174	SYSTEM	175	LPRINT
176	DEF	177	POKE
178	PRINT	179	CONT
180	LIST	181	LLIST
182	DELETE	183	AUTO
184	CLEAR	185	CLOAD
186	CSAVE	187	NEW
188	TAB(	189	TO
190	FN	191	USING
192	VARPTR	193	USR
194	ERL	195	ERR
196	STRING$	197	INSTR
198	CHECK	199	TIME$
200	MEM	201	INKEY$
202	THEN	203	NOT
204	STEP	205	+
206	-	207	*
208	/	209	[
210	AND	211	OR
212	>	213	=
214	<	215	SGN
216	INT	217	ABS
218	FRE	219	INP

220	⁄⁄	POS	221	‖	SQR
222	⑊	RND	223	╱	LOG
224	◣	EXP	225	╿	COS
226	◤	SIN	227	╲	TAN
228	◥	ATN	229	▼	PEEK
230	●	CVI	231	Ψ	CVS
232	↑	CVD	233	⊡	EOF
234	⋱	LOC	235	O	LOF
236	┌	MKI$	237	┐	MKS$
238	◖	MKD$	239	└	CINT
240	┘	CSNG	241	╳	CDBL
242	◆	FIX	243	✲	LEN
244	◓	STR$	245	◗	VAL
246	·	ASC	247	∷	CHR$
248	□	LEFT$	249	⁙	RIGHT$
250	⌘	MID$	251	↓	'

Colour-Basic Doppeltokens

255	128	┓	COLOUR	255	129	┏	FCOLOUR
255	130	┛	KEYPAD	255	131	┗	JOY
255	132	┫	PLOT	255	133	┣	FGR
255	134	┻	LGR	255	135	┳	FCLS
255	136	━	PLAY	255	137	⎸	CIRCLE
255	138	⎹	SCALE	255	139	‾	SHAPE
255	140	_	NSHAPE	255	141	□	XSHAPE
255	142	▌	PAINT	255	143	·	CPOINT
255	144	▐	NPLOT	255	145	▬	SOUND
255	146	┏	CHAR	255	147	┓	RENUM
255	148	≻	SWAP	255	149	≺	FKEY
255	150	▶	CALL	255	151	◀	VERIFY
255	152	▪	BGRD	255	153	▮	NBGRD

(Gueltig fuer die neuen ROMs)

Anhang B: Dezimal-, Hexadezimal-, ASCII-Tabelle

0	= 00H		1	= 01H		2	= 02H	
3	= 03H		4	= 04H		5	= 05H	
6	= 06H		7	= 07H		8	= 08H	
9	= 09H		10	= 0AH		11	= 0BH	
12	= 0CH		13	= 0DH		14	= 0EH	
15	= 0FH		16	= 10H		17	= 11H	
18	= 12H		19	= 13H		20	= 14H	
21	= 15H		22	= 16H		23	= 17H	
24	= 18H		25	= 19H		26	= 1AH	
27	= 1BH		28	= 1CH		29	= 1DH	
30	= 1EH		31	= 1FH		32	= 20H	
33	= 21H	!	34	= 22H	"	35	= 23H	#
36	= 24H	$	37	= 25H	%	38	= 26H	&
39	= 27H	'	40	= 28H	(	41	= 29H	)
42	= 2AH	*	43	= 2BH	+	44	= 2CH	,
45	= 2DH	-	46	= 2EH		47	= 2FH	/
48	= 30H	0	49	= 31H	1	50	= 32H	2
51	= 33H	3	52	= 34H	4	53	= 35H	5
54	= 36H	6	55	= 37H	7	56	= 38H	8
57	= 39H	9	58	= 3AH		59	= 3BH	,
60	= 3CH	<	61	= 3DH	=	62	= 3EH	>
63	= 3FH	?	64	= 40H	@	65	= 41H	A
66	= 42H	B	67	= 43H	C	68	= 44H	D
69	= 45H	E	70	= 46H	F	71	= 47H	G
72	= 48H	H	73	= 49H	I	74	= 4AH	J
75	= 4BH	K	76	= 4CH	L	77	= 4DH	M
78	= 4EH	N	79	= 4FH	O	80	= 50H	P
81	= 51H	Q	82	= 52H	R	83	= 53H	S
84	= 54H	T	85	= 55H	U	86	= 56H	V
87	= 57H	W	88	= 58H	X	89	= 59H	Y
90	= 5AH	Z	91	= 5BH	[	92	= 5CH	\
93	= 5DH	]	94	= 5EH	^	95	= 5FH	_
96	= 60H	`	97	= 61H	a	98	= 62H	b
99	= 63H	c	100	= 64H	d	101	= 65H	e
102	= 66H	f	103	= 67H	g	104	= 68H	h
105	= 69H	i	106	= 6AH	j	107	= 6BH	k
108	= 6CH	l	109	= 6DH	m	110	= 6EH	n
111	= 6FH	o	112	= 70H	p	113	= 71H	q
114	= 72H	r	115	= 73H	s	116	= 74H	t
117	= 75H	u	118	= 76H	v	119	= 77H	w
120	= 78H	x	121	= 79H	y	122	= 7AH	z
123	= 7BH	{	124	= 7CH	¦	125	= 7DH	}
126	= 7EH	~	127	= 7FH	█	128	= 80H	┓
129	= 81H	┏	130	= 82H	┛	131	= 83H	┗
132	= 84H	┫	133	= 85H	┣	134	= 86H	┻
135	= 87H	┳	136	= 88H	━	137	= 89H	┃

138	=	8AH	|	139	=	8BH	‾	140	=	8CH	_
141	=	8DH	□	142	=	8EH		143	=	8FH	·
144	=	90H		145	=	91H		146	=	92H	
147	=	93H		148	=	94H		149	=	95H	
150	=	96H		151	=	97H		152	=	98H	
153	=	99H		154	=	9AH		155	=	9BH	
156	=	9CH		157	=	9DH		158	=	9EH	
159	=	9FH		160	=	A0H		161	=	A1H	
162	=	A2H		163	=	A3H		164	=	A4H	
165	=	A5H		166	=	A6H		167	=	A7H	
168	=	A8H		169	=	A9H		170	=	AAH	
171	=	ABH		172	=	ACH		173	=	ADH	
174	=	AEH		175	=	AFH		176	=	B0H	
177	=	B1H		178	=	B2H		179	=	B3H	
180	=	B4H		181	=	B5H		182	=	B6H	
183	=	B7H		184	=	B8H		185	=	B9H	
186	=	BAH		187	=	BBH		188	=	BCH	
189	=	BDH		190	=	BEH		191	=	BFH	
192	=	C0H		193	=	C1H	≡	194	=	C2H	+
195	=	C3H		196	=	C4H		197	=	C5H	×
198	=	C6H	♥	199	=	C7H		200	=	C8H	♠
201	=	C9H	♣	202	=	CAH	■	203	=	CBH	
204	=	CCH		205	=	CDH	◢	206	=	CEH	
207	=	CFH		208	=	D0H		209	=	D1H	
210	=	D2H		211	=	D3H		212	=	D4H	
213	=	D5H		214	=	D6H		215	=	D7H	≡
216	=	D8H		217	=	D9H	—	218	=	DAH	_
219	=	DBH		220	=	DCH	//	221	=	DDH	
222	=	DEH	\\\\	223	=	DFH	/	224	=	E0H	◣
225	=	E1H		226	=	E2H		227	=	E3H	\\
228	=	E4H	◥	229	=	E5H		230	=	E6H	●
231	=	E7H		232	=	E8H	↑	233	=	E9H	
234	=	EAH		235	=	EBH	O	236	=	ECH	
237	=	EDH		238	=	EEH		239	=	EFH	
240	=	F0H		241	=	F1H		242	=	F2H	♦
243	=	F3H		244	=	F4H		245	=	F5H	
246	=	F6H	·	247	=	F7H	::	248	=	F8H	□
249	=	F9H		250	=	FAH		251	=	FBH	↓
252	=	FCH	:	253	=	FDH	←	254	=	FEH	
255	=	FFH	→								

Anhang C <u>Der Farbspeicher</u>

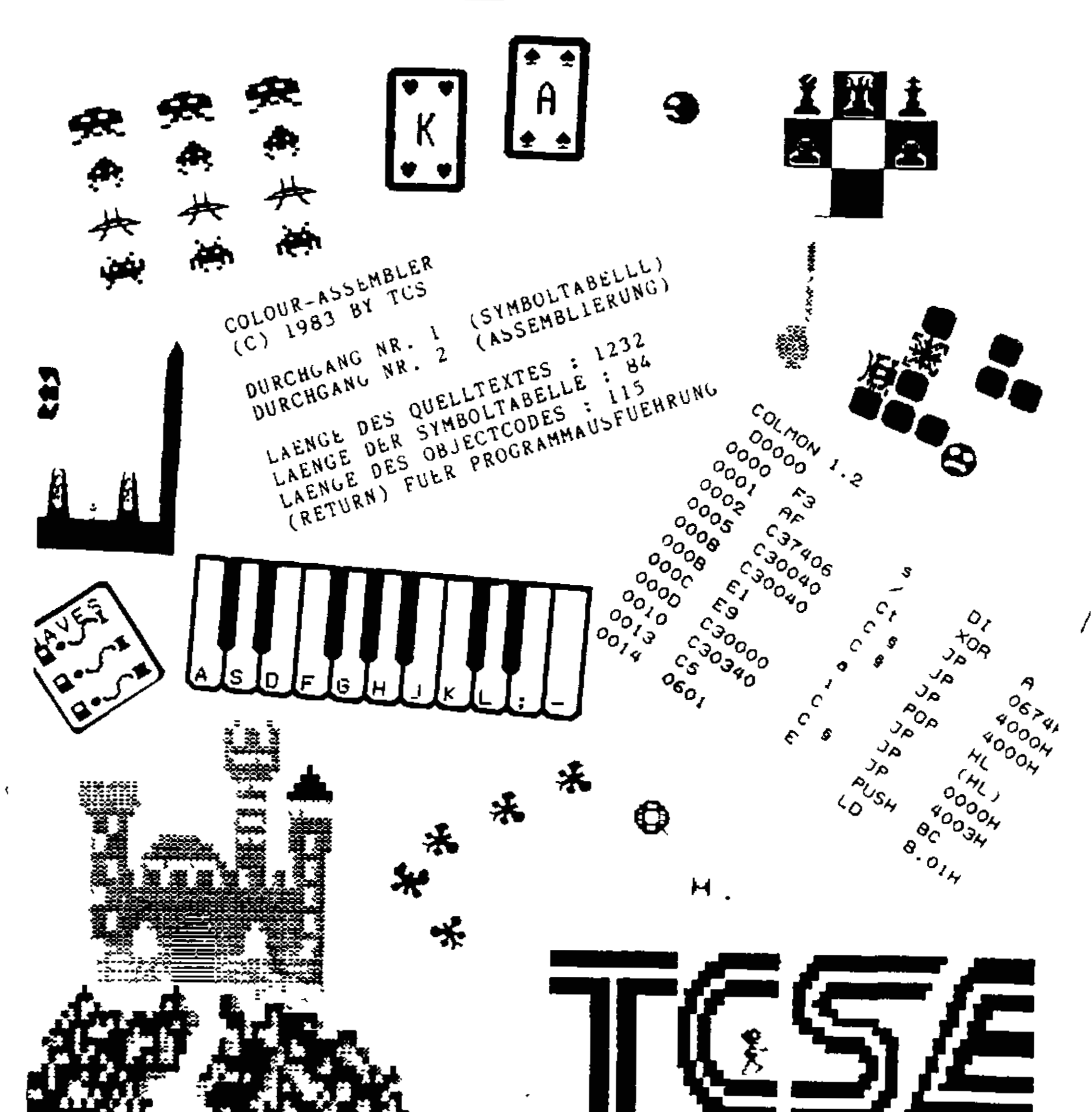
Vielleicht ist Ihnen auch schon aufgefallen, daß man beim POKEn in den Farbspeicher bei gleichem Argument andere Farben bekommt als mit dem COLOUR-Befehl.
Folgende Tabelle gibt Ihnen an, wie die einzelnen Farben abgespeichert werden.
(Hinweis: Der Farbspeicher hat nur 4 Bit pro Adresse, d.h. wenn Sie den Farbspeicher mit dem PEEK-Befehl abfragen müssen Sie das Ergebnis mit 15 'undieren', um die 4 nicht belegten Bits zu unterdrücken. Z.B.:
X=PEEK(&HF000) AND 15

Es folgt nun die Vorbelegung der Farbtabelle ab &H4390 :

Farbe	Bit 3	Bit 2	Bit 1	Bit 0	Wert	Farb-Nr.
Grau	0	0	0	0	0	10
Cyan	0	0	0	1	1	7
Rot	0	0	1	0	2	3
Weiß	0	0	1	1	3	1
Gelb	0	1	0	0	4	4
Grün	0	1	0	1	5	2
Orange	0	1	1	0	6	5
Hellgelb	0	1	1	1	7	11
Blau	1	0	0	0	8	6
Hellblau	1	0	0	1	9	9
Rotviolett	1	0	1	0	10	15
Blauviolett	1	0	1	1	11	12
Hellgrau	1	1	0	0	12	13
Türkis	1	1	0	1	13	14
Magenta	1	1	1	0	14	8
Hellweiß	1	1	1	1	15	16

Colour-Genie

Software



****
Verzeichnis der verfugbaren Colour-Genie-Software
****

1.) HILFSPROGRAMME, PROGRAMMIERSPRACHEN

COLOUR-COMPILER: Dieses Programm ist extrem wertvoll. Es bietet die Moglichkeit, Basicprogramme in Maschinensprācheprogramme zu ubersetzen.
Der wesentliche Vorteil: Ihr Programm lauft nach Compilierung c.a. 40 mal (!) schneller.
Dies geschieht interaktiv, d.h. Basicprogramm, Maschinenprogramm und Compiler stehen gleichzeitig im Speicher, was ein sehr schnelles Arbeiten ermoglicht. Dabei konnen c.a. 9.5 kByte Basicprogramm verarbeitet werden. Verzichtet man auf die hochauflosende Grafik, steht noch mehr Speicher zur Verfugung.
Voraussetzung: Ihr COLOUR-GENIE MUSS AUF 32K RAM ERWEITERT SEIN. Die meisten Colour-Basic-Befehle werden vom Colour-Compiler unterstutzt. Die wesentlichste Einschrankung ist, daß der Colour-Compiler nur Ganzzahlen (Integers) verarbeitet.
Preis: 69.- DM

COLOUR-MONITOR I: Ein Maschinensprachemonitor mit starkem Befehlssatz, z.B. Disassemlieren, Ascii/Hex-Dump, Bander laden/schreiben, Bytes suchen, Speicher editieren/verschieben/relozieren, Hex/Dezimal-Umwandlung u.a.m.
Preis: 39.- DM

COLOUR-ASSEMBLER: Mit diesem Programm konnen Maschinensprācheprogramme auf komfortable Weise entwickelt werden. Das Programmieren geschieht mit Z80-Mnemonics, Labels etc.
Der Assembler erzeugt dann das Maschinenspracheprogramm, das zur Probe auch direkt ausgefuhrt werden kann.
Eine weitere Besonderheit des Assemblers ist, daß der Assembler-Text im ganz normalen Basic-Modus erzeugt wird, so daß alle Basic-Befehle wie EDIT, LLIST, CSAVE vom Assembler ausgenutzt werden.
Preis: 69.- DM

SOUND-EDITOR (auch im Handbuch "Colour Basic leicht gelernt" aufgelistet) : Ein nutzliches Hilfsprogramm zur Programmierung des PSG-ICs (Sound-Chip). Sie editieren die PSG-Register auf dem Bildschirm und der entsprechende Ton wird gleichzeitig ausgegeben. Abschließend gibt das Programm alle PSG-Registerinhalte in dezimaler Schreibweise aus.
Preis: 25.- DM

ZEICHEN-EDITOR (auch im Handbuch "Colour Basic leicht gelernt" aufgelistet) : Dieses Programm ermoglicht es, 64 der 128 frei definierbaren Zeichen auf dem Bildschirm ubersichtlich zu editieren. Es konnen auch Grafiken, wie z.B eine Schreibschrift, erstellt werden. Abschließend werden die definierten Zeichen in ein Basicprogramm geschrieben, was diese fur den spateren Gebrauch oder zur Entwichlung eigener Programme wieder definiert. So wird die umstandliche Handhabung der Programmierung der definierbaren Zeichen umgangen.
Preis: 25.- DM

GRAFIK-EDITOR: Wenn Sie Ihr Colour-Genie auf 32K RAM aufgerustet haben, konnen Sie dieses Programm einsetzten, denn der "Grafik-Editor" hat eine Lange von 26000 Bytes!
Er bietet die Moglichkeit, Grafiken im FGR-Modus mit einer Vielzahl von leistungsstarken Kommandos zu kreieren. Abschließend wird ein Basic-Programm erzeugt werden, das, eingebunden in ein eigenes Programm, das Bild blitzschnell wieder auf den Bildschirm bringt. Sie konnen also auch komplexe Grafiken sehr schnell fertigstellen und sie hinterher beliebig verwenden.
Preis: 69.- DM

SHAPER: Die Handhabung des Basicbefehls "SHAPE" ist recht umstandlich. Hier bringt der "Shaper" Abhilfe: Mit wenigen Tastendrucken erzeugen Sie eine Figur, die abschließend im Foramt der "Shape-Table" abgespeichert wird.
Preis: 25.- DM

COLROT: Ein sehr nutzliches Grafikprogramm, für alle die mit großen Texten Aufmerksamkeit erregen wollen (z.B. im Schaufenster). Colrot erzeugt Laufschriften mit bis zu 512 Zeichen Lange, Titel und Fußschrift, Inversdarstellung, Intermezzo u.v.a.m. Neu:Texte konnen auf Band gespeichert werden. Ein komfortabler Editor ermoglicht einfaches Arbeiten.
Preis: 69.- DM

BASICODE 2 (!!): Der neue Basicode-Standard fur das Colour-Genie (kompatibel zu Basicode 1). Dieses Programm macht ihr Colour-Genie Basicode kompatibel !
Basicode ist das Standard-Basic, in dem z.B. das WDR-Fernsehen Programme austrahlt. Eine Hardwareanderung ist nicht nötig.
Preis: 25.- DM

ZEICHENEDITOR+: Dieser Editor, in Maschinensprache geschrieben, ermoglicht es alle 128 Zeichen sehr schnell und komfortabel zu editieren. Dabei gibt es sogar Befehle um Kreise zu zeichnen, Zeichen zu kopieren/duplizieren/rotieren/invertieren u.v.a.m.
Preis: 39.- DM

BASIC+5: Dieses Programm erweitert das normale Basic um 5 Befehle: Im Grafikmodus konnen beliebige Texte dargestellt werden, Rechteckflachen konnen gemalt werden, mit "SAVE" konnen Machinenspracheprogramme gesichert und mit "LOAD" komfortabel geladen werden. Ferner steht ein Eingabebefehl zur Verfugung, der an beliebige Speicherzellen schreibt.
Preis: 39.- DM

ZEICHENSATZE: Dieses Programm laßt Sie den vorhandenen Zeichensatz fur alle ASCII-Zeichen wahlweise gegen einen von acht neuen, interessanten Zeichensatzen ersetzen.
Fur dieses Programm muß Ihr Colour-Genie mit 32K RAM ausgerustet sein.
Preis: 25.- DM

SCREEN-PRINTER: (Auch in dem Buch "Das Colour-Genie-Buch 1" aufgelistet.) Auf dieses Programm haben die Besitzer des STAR-Druckers DP 510 bzw. DP 515 sicher gewartet !
Es ermoglicht es Ihnen, den Bildschirm jederzeit komplett auf Ihren Drucker auszugeben, egal ob Sie im FGR- oder im LGR-Modus sind. Definierte und feste Grafikzeichen werden ebenfalls mit ausgedruckt. Auch fur dieses Programm brauchen Sie 32K RAM und naturlich einen STAR DP 510 / DP 515 Drucker.
Preis: 39.- DM

COLOUR-MONITOR III: Dieses Programm wird alle, die auf Maschinensprache-Ebene programmieren begeistern !
Der "Colour-Monitor III" ist nicht nur ein Monitor mit den ublichen Befehlen, wie ASCII-Dump, Hex-Dump, Disassemblieren, Bander schreiben etc., sondern ein Monitor mit eingebautem Assembler, Find-Assembler und der Moglichkeit Maschinenspracheprogramme in Einzelschrittausführung zu testen ! Das Arbeiten wird neben dem Super-Befehlssatz durch raffinierte Aufteilung des Bildschirms erleichtert.
Fur dieses Programm benötigen Sie ein Colour-Genie mit 32K Ram.
Ein ausfuhrliches deutsches Handbuch liegt bei.
Preis: 95.- DM

COLOUR-FORTH: Endlich gibt es eine leistungsstarke Version der Forth-Programmiersprache auch fur das Colour-Genie!
"Colour-Forth" ist eine fur Kassettenbetrieb angepaßte Version des FIG-Forth, die um viele Befehle fur das Colour-Genie erweitert wurde (Graphik, Ton...). Der wesentliche Vorteil der Programmiersprache Forth ist die extrem schnelle Ausfuhrungszeit der Programme, die in unserem Forth noch durch Benutzung des eingebauten Assemblers gesteigert werden kann ! Für dieses Programm benotigen Sie 32K Ram, sowie die neuen Basic-Roms, die seit April '83 in alle Colour-Genies eingebaut sind. Ein ausführliches deutsches Handbuch wird mitgeliefert.
Preis: 95.- DM

COLOUR-TEXT: Endlich konnen alle Colour-Genie-Besitzer, die einen Drucker haben in den Genuß einer komfortablen Textverarbeitung kommen ! Colour-Text erweitert die Bildschirmbreite auf 55 Zeichen, die maximale Zeilenlange ist 64 Zeichen (horizontales Scrolling). Sie sind mit dem Cursor standig im Text, und keine Steuerzeichen verunstalten den Text, da die verschiedenen Schriftarten durch Farben gekennzeichnet sind. (Fur Benutzer eines Schwarz/Weiss-Sichtgerates extra Statuszeile.) Das Programm unterstutzt alle gangigen Drucker (STAR, EPSON, ITOH, Typenrad). Wenn Sie einen grafikfahigen STAR-, EPSON- oder ITOH-Drucker haben konnen Sie sogar selbstdefinierte Grafikzeichen in den Text einfugen !
Ein deutsches Handbuch liegt bei.
Preis: 95.- DM

2.) SPIEL- u. GRAFIKPROGRAMME

INVASION AUS DEM WELTRAUM: Ein Maschinenspracheprogramm, das die Sound- u. Grafikmoglichkeiten des Colour-Genies voll ausnutzt. Ihre Aufgabe ist es, einen Pulk von Invasoren abzuwehren, der sich der Erde nahert.
Die Invasoren werden dabei immer schneller und gefahrlicher.
Preis: 39.- DM

PUNKTEJAGD: Bei diesem Spiel geht es darum, alle Punkte in einem Labyrint aufzusammeln, bevor Sie von einem Wachter eingeholt werden. Schnelle Grafik und Ton durch Maschinensprache.
Preis: 25.- DM

WURM: Unser derzeit schnellstes Action-Spiel. Ein Wurm kommt von oben den Bildschirm herab und versucht Sie zu vernichten. Dabei hat er die Spinne, die Fliege und die Ente als Helfer. Kein Spiel fur ruhige Stunden!
Preis: 39.- DM

BREAK OUT: Bei diesem Video-Spiel muß eine Mauer mit Ihrem Ball zerstört werden. Dabei konnen verschieden Schwierigkeitsgrade vorgewahlt werden. Hohe Geschwindigkeit durch Maschinenspracheprogrammierung.
Preis: 39.- DM

DEMOPROGRAMM: Hiermit konnen Sie besser als mit der mit Ihrem Colour-Genie z.Zt. mitgelieferten englischen Demokassette die Moglichkeiten Ihres Colour-Genies demonstrieren.
Preis: 25.- DM

ANDROMEDA: Ein erstklassiges dreidimensionales Weltraumspiel mit ansprechender Grafik. Feindliche Raumschiffe kommen auf Sie zu; vernichten Sie diese, bevor sie Ihnen wertvolle Energie abgesaugt haben. Ist die Zeit abgelaufen, so eilt Ihnen Ihre Mutterbasis zu Hilfe; aber das Auftanken will auch gelernt sein.
Preis: 39.- DM

MAU-MAU: Endlich haben Sie einen Spielpartner, der nicht wutend die Karten wegwirft, wenn er am verlieren ist. Dies durfte wohl aber auch selten der Fall sein, vielmehr wahrscheinlich ist es, daß Sie den Computer vor Wut ausschalten, da ihr Computer Dank einer hervorragenden Taktik auf Sieg programmiert ist.
(Er schummelt nicht !!) Gespielt wird nach den ublichen Regeln.
Preis: 25.- DM

HEKTIK: Stellen Sie sich vor, Sie wären in einem Neubau mit 6 Geschossen, bei dem die Ebenen durch Leitern verbunden sind. Jetzt kommen Ihnen von oben Verfolger entgegen. Sie mußen sich ihnen stellen, denn es gibt keinen Fluchtweg. Graben Sie an strategisch wichtigen Stellen Locher in den Boden, um so Ihre Verfolger auszuschalten.
Preis: 39.- DM

METEOR: Ein Super-Action-Spiel im Grafik-Modus Ihres ColourGenies. Sie befinden sich in einem Meteoritenfeld und werden von Meteoriten, Sternen und schiessenden Ufos bedrängt. Das Programm zeichnet sich durch sehr schnelle Grafik und gute Toneffekte aus.
Preis: 69.- DM

MOTTEN: Bei diesem Videospiel müßen Sie Kolonnen von Motten bekämpfen, die in gefährlichen Sturzflügen anfliegen und dabei auch noch schiessen.
Preis: 39.- DM

PANIK: Ein Programm für 2 Spieler. Man muß versuchen, den Gegner einzumauern. Wählbare Geschwindigkeit macht "Mauer" entweder zu einem Reaktions- oder zu einem Strategiespiel. Das Programm kann wahlweise mit Joysticks oder mit der Tastatur bedient werden.
Preis: 25.- DM

TAUSENDFUß: Hier sollen Sie einen Wurm so steuern, daß er nur Futter aber kein Gift frißt. Dazu kommen noch Kraftfutter und Gegengift. Ein sehr unterhaltsames Geschicklichkeitsspiel für die ganze Familie. Bei Spielbeginn können viele verschiedene Geschwindigkeiten und Schwierigkeitsstufen vorgewählt werden.
Preis: 39.- DM

COLOUR-SCHACH: Jetzt können Sie gegen Ihr Colour-Genie auch Schach spielen. Colour-Schach bietet 4 verschiedene Spiel-stärken, Aufzeichnung eines laufenden Spieles auf Kassette, Ändern von Stellungen, einen Demonstrationsmodus und anderes mehr. Die Figuren werden auf ansprechende Weise grafisch dargestellt.
Preis: 69.- DM

EXNIMROID: Ein Denkspiel, abgeleitet vom bekannten Nim-Spiel. Es geht darum, aus mehreren Häufchen den letzten Spielstein zu nehmen. Sie spielen gegen den Computer.
Preis: 25.- DM

KINGS: Eine Regierungssimulation, bei der Sie für eine Dauer von 8 Jahren eine Insel regieren sollen. Dabei kann nur der kluge Einsatz aller Faktoren zu einem Erfolg führen. Haben Sie Ihre Amtszeit überlebt, werden Ihre Taten ausgewerted, und Sie können dann auch weiter regieren.
Preis: 25.- DM

CONQUER: Ein spannendes Videospiel - erobern Sie die Spinnenfestung !
Preis: 25.- DM

MAMPFMAN: Ahnlich wie unser Spiel "Punktejagd", nur daß Sie hier von mehreren Gespenstern verfolgt werden, die Sie unter bestimmten Bedingungen jedoch auch fressen konnen.
Preis: 25.- DM

MAMPFMAN 2: Die neue Version von Mampfman ! Excellente Grafik und hohe Spieldynamik. Ein sehr empfehlenswertes Programm !
Preis: 39.- DM

COLOUR-KONG: Das absolute Videospiel ! Retten Sie Ihre Freundin vor dem wilden Affen. Mehrere Ebenen, super Grafik und Musik. Auf dem Band finden Sie Versionen fur 32K und 16K RAM
Preis: 69.- DM

PANZERKAMPF: Ein Spiel fur zwei Personen (JOYSTICKS und 32K RAM erforderlich). Liefern Sie sich mit Ihrem Gegner spannende Panzerschlachten ! Das Programm verfugt uber 3 verschiedene Gelande mit Minen etc.
Preis: 69.- DM

EIS: Ein spannendes, zugleich jedoch auch recht schwieriges Actionspiel mit strategischen Elementen. Sie befinden sich in einem Kuhlraum und mußen Eisblocke so ans Rutschen bringen, daß die gefahrlichen Schneemonster vernichtet werden.
Preis: 39.- DM

EAGLE: Vernichten Sie verschiedene Schwadronen von Vögeln, bevor Sie versuchen das große Mutterschiff zu zerstoren.
Preis: 39.- DM

GAME of LIFE: Dieses bekannte Programm simuliert das Wachstum von Bakterienkulturen. Die Regeln werden im Programm erklart und einige interessante Figuren sind fest abgespeichert.
Preis: 25.- DM

SAUG: Ein lustiges Videospiel, bei dem Sie aus einem unterirdischen Labyrint Termiten ansaugen mußen, wobei Sie sich vor giftigen, roten Termiten huten mußen.
Das Spiel kann auch mit Joysticks gesteuert werden.
Preis: 39.- DM

ExREVERSIC: Spielen Sie Reversi gegen Ihr Colour-Genie ! Dabei kann zwischen mehreren Spielstufen gewahlt werden. Das Spielbrett wird auf ansprechende Weise graphisch dargestellt.
Preis: 39.- DM

BANG-BANG: Zwei Cowboys begegnen sich in der Prarie und liefern sich ein erbittertes Duell. Fur dieses Spiel sind Joysticks erforderlich.
Preis: 39.- DM

NETZO: Versuchen Sie mit Ihrem Pinsel alle Flachen auszumalen, ohne von Ihren Gegnern erwischt zu werden.
Preis: 39.- DM

DEATH-TRAP, ein dreidimensionales Grafik-Abenteuerspiel (Adventure):
Ein Programm zum wahnsinnig werden! Sie befinden sich in einem Labyrinth mit uber 1100 Raumen, in dem es von gefährlichen Einwohnern und Gegenstanden wimmelt. Finden Sie die wichtigen Gegenstande und wenden Sie sie richtig an, um aus dem Labyrinth zu entkommen. Zwei Vorraussetzungen mussen allerdings erfullt sein: Ihr Colour-Genie muß 32K RAM haben, und Sie mussen elementare Englischkenntnisse besitzen, da die Sie Kommandos in Form von englischen Satzen eingeben.
Preis: 69.- DM

TCS-CHOPPER: Die Geschmacker sind verschieden - aber dies ist wohl das beste Videospiel, das es z.Zt. fur das Colour-Genie gibt! Dreizehn Ihrer Kameraden sind in einem von vier riesigen Labyrinthen ausgesetzt worden. Versuchen Sie nun, diese mit Ihrem Hubschrauber zu retten! Aber Vorsicht, Ihre Freunde werden scharf bewacht!
32K RAM sind auch hier notwendig.
Preis: 69.- DM

DOPPEL-WURM: Vielleicht kennen Sie unser beliebtes Spiel "Tausendfuß". "Doppel-Wurm" ist sehr ahnlich, allerdings spielen hier zwei Spieler gegeneinander, was die Spannung naturlich erhöht. Achtung: Nur fur Joystick-Besitzer !
Preis: 39.- DM

MADTREE: Ein schwieriges Videospiel: Sie sind eine fleißige Biene und mussen eine Reihe von Blumen standig bestauben, damit diese nicht eingehen. Doch diese Blumen sind undankbar und gefahrden Sie.
Preis: 39.- DM

DIG-BOY: In Ihrem unterirdischen Reich kampfen Sie gegen Drachen und andere Untiere. Locken Sie diese unter einen der wackligen Felsen, um sie zu zerschmettern.
Dieses Spiel ist sehr empfehlenswert, da es eine Menge Abwechslung bietet.
Preis: 39.- DM

EMPIRE: Fur alle, die nicht nur reine Aktion wollen, ist "Empire" das ideale Spiel. Dieses Spiel kann mit bis zu sechs Spielern gleichzeitig gespielt werden, der Computer spielt ebenfalls mit. Jeder Spieler hat ein kleines Reich, das nun durch Handel, Politik, Krieg u.s.w. zu Wohlstand kommen soll. Dabei stehen Sie mit den anderen Nationen standig in Beziehung, sei es durch Handel oder durch Krieg.
Ansprechende Grafik macht das Programm noch interessanter.
Ihr Colour-Genie muß fur "Empire" auf 32K RAM aufgerustet sein.
Preis: 69.- DM

LUNAR-LANDER: Eine perfekte Mondlandungssimulation mit Supergrafik ! Die Idee ist zwar alt, aber die Ausfuhrung dieses Programmes macht den "Lunar-Lander" zu einem unserer besten Spiele. Sie benotigen 32K Ram. Das Spiel kann wahlweise auch mit den EG2013-Joysticks gesteuert werden.
Preis: 39.- DM

COLOUR-FROGGER: Retten Sie Ihren Frosch zuerst uber die Straße und dann uber den Fluß ! Ein spannendes Unterhaltungsspiel, an dem die ganze Familie lange Zeit Spass haben wird. Besonders gelungen ist bei diesem Programm die musiklische Untermalung.
Preis: 39.- DM

CRAZY-PAINT: Das neueste Spiel von J. Buchmuller, dem Author des TCS-Choppers - Super-Grafik und Super-Ton !
In 16 Levels mussen Sie, ahnlich wie bei unserem Spiel 'Netzo', versuchen alle Flachen auszumalen, ohne sich von den hektischen Gegnern erwischen zu lassen.
Preis: 69.- DM

MEMORY: Das beliebte Brettspiel nun auch als Version fur das Colour-Genie !
Schone Grafik und die gute Spielidee machen dieses Programm empfehlenswert.
Preis: 39.- DM

3.) LERNPROGRAMME, ELEKTRONIKPROGRAMME:

MATHEMATIK-LERNPROGRAMM: Interessant für Kinder bis in's Grundschulalter. Wahlweise können die vier Grundrechenarten geübt werden, wobei man zwischen verschiedenen Schwierigkeitsgraden wählen kann. Lustige grafische Gestaltung sorgt dafür, daß das Üben Spass macht.
Preis: 25.- DM

GRUNDEIGENSCHAFTEN DES LICHTS (OPTIK): Diese 32KByte lange Programm erklärt die Brechung, Reflexion, verschiedene Linsentypen... Ansprechende Grafik, Benutzereingaben, Menu und Übungsaufgaben machen dieses Programm sehr lehrreich.
Preis: 39.- DM

NE555: Erklärung und Anwendung des Timer-ICs NE555 werden mit diesem Programm zum Kinderspiel. Tolle Grafik und ausführliche Dokumentation heben dieses Programm hervor.
Preis: 25.- DM

PLOTTER: Plotter ist ein anspruchsvolles Mathemtikprogramm, das beliebige Funktionen zeichnet, Wertetabellen erstellt ...
Preis: 39.- DM

PHYSIK DES TRANSISTORS: Ähnlich aufwendig wie beim Programm 'Optik' werden Sie hier in die Grundeigenschaften des Transistors, verschiedene Schaltungen, Kennlinien u.s.w. eingewiesen.
Für dieses Programm sind 32K Byte RAM erforderlich.
Preis: 39.- DM

4.) MUSIKPROGRAMME

MUSIK: Dieses Programm spielt festprogrammierte Musikstucke mit verbluffender Qualitat.
Preis: 25.- DM

ORGEL: Spielen Sie Orgel auf Ihrem Colour-Genie !
Ihnen stehen zwei Manuale, einstellbares Delay und sogar eine Schlagzeugbegleitung zur Seite.
Preis: 25.- DM

COLOUR-SYNTHESIZER: Ein Programm, das man gesehen haben muß, um es uberhaupt fur moglich zu halten. Der Colour-Synthesizer macht aus Ihrem Colour-Genie einen vollwertigen 3-Kanal Synthesizer mit VCO, VCA, Hullkurve, Schlagzeug...
Acht Einstellungen aller Regler konnen vorprogrammiert, auf Tastendruck abgerufen und auf Band gesichert werden.
Verbluffend ist auch die graphische Darstellung.
Preis: 69.- DM

COMPOSER: Mit diesem Programm konnen Sie sehr einfach eigene Musikstucke komponieren. Noten, Pausenzeichen u.s.w. werden grafisch dargestellt, konnen editiert und auf Band gespeichert werden.
Preis: 39.- DM

** STAND DER TCS-SOFTWARELISTE, COLOUR-GENIE, VOM 18.12.1983 **

Beachten Sie auch die fur das Colour-Genie lieferbaren Bucher:
Technisches-Handbuch: Enthalt die kompletten Schaltplane, Erklarungen, CRTC-Unterlagen, PSG-Unterlagen...
Preis: 49.- DM

Dokumentiertes ROM-Listing fur das Colour-Genie: Unerlaßlich fur den Maschinensprache-Programmierer. In diesem Buch ist der komplette ROM und reservierte RAM aufgelistet und erklart.
Preis: 59.- DM

DAS Colour-Genie BUCH I: Auf uber 100 Seiten finden Sie hier interessante Programme - von Spielen bis zu Hilfsprogrammen, zum Teil auch in Maschinensprache - sowie wertvolle Tricks und Erklarungen. Ein Muß fur jeden Colour-Genie-Besitzer!
Preis: 39.- DM

In Vorbereitung: Software fur Diskette, neue Spiele ...

5.) PROGRAMME FUR DISKETTENBENUTZER

TAPE-DISK-CONVERTER: Dieses Programm ermoglicht es Ihnen, Ihre Maschinensprache-Kassetten (sofern ungeschutzt) auf Diskette zu uberspielen, wobei eine Verschiebung im Speicher moglich ist. Außerdem kann man von Diskette auf Diskette, von Kassette auf Kassette und von Diskette auf Kassette uberspielen.
Preis: 45.- DM

COLZAP: Mit "Colzap" konnen Sie auf jeder (ungeschutzten) Colour-Genie-Diskette alle Sektoren lesen, beliebig andern (in ASCII oder Hex), Sektoren schreiben, den Disknamen anzeigen, den freien Platz auf der Diskette ausgeben lassen, u.a.m
Preis: 45.- DM

BESTELLFORMULAR

Zur Bestellung von Colour-Genie-Programmen konnen Sie dieses Formular benutzten. Schicken Sie dieses an Ihren GENIE-Handler oder direkt an

TCS Computer GmbH
Kolnstraße 4
5205 St. Augustin 2
Tel.: 02241 / 28071

Bankverbindung:
Raiffeisenbank Menden BLZ 37069507 Konto-Nr. 1525

Der Versand erfolgt per Nachnahme oder Vorauskasse auf obiges Konto (dies naturlich nicht, wenn Sie bei Ihrem Handler bestellen).

Ich bitte um Zusendung folgender TCS Colour-Genie-Programme:

Anzahl	Name des Programms	Preis/DM	Gesamt

Gesamt: DM

Meine Adresse ist:

Name: ....

Straße: ....

Ort: ....

Datum: .... Unterschrift: ....

Raum fur Notizen

Kalle Braun · Jürgen Buchmüller · Frank Seger

Das Colour-Genie

Buch 3



Das Colour-Genie Buch 3

ISBN 3 88965 007 4

Vorwort

Aller guten Dinge sind drei!

Hier also das dritte Colour-Genie Buch. Wieder einmal hatten wir Zeit und einen freien Rechner, so entstand das

»Colour-Genie Buch III«

Wir hoffen, daß auch dieses Buch Ihnen wieder viele Anregungen vermittelt und Sie viel Freude an den gelisteten Programmen haben werden.

Unter anderem finden Sie in diesem Buch viele Programme für Diskettenbesitzer, so zum Beispiel ein Programm zum alphabetischen Sortieren des Inhaltsverzeichnisses. Auch die Spiele kommen wieder nicht zu kurz. Wieder hat Jürgen Buchmüller extra für das Buch in die Tasten, will sagen, das Keyboard gegriffen und heraus kam ,,Ballons''.

Da die in diesem Buch gelisteten Programme doch schon relativ schwierig sind, empfehlen wir Ihnen, in jedem Fall zuerst die Colour-Genie-Bücher I und II durchzuarbeiten.

Wir hoffen, daß Sie auch an diesem Buch wieder Ihre Freude haben werden!

Bonn, im Mai 1984

Kalle Braun
Jürgen Buchmüller
Frank Seger

Inhaltsverzeichnis

Allgemeine Hinweise

Bitte beachten Sie, daß die Zeichen '§' und 'A' in den Listings den Zeichen '@' bzw. '[' auf Ihrer Tastatur entsprechen.

Die meisten Basic-Programme in diesem Buch sind 'entpackt', d.h. nach jedem Befehl folgt ein Leerzeichen, in jeder Zeile steht nur ein Befehl usw. Es ist naturlich nicht notig, alle diese Leerzeichen mit einzugeben, es erleichtert jedoch das Programmverstandnis und die Fehlersuche.

Falls Sie sich tiefer in die Maschinensprache Ihres Colour-Genie einarbeiten wollen, empfehlen wir Ihnen

'Programmierung des Z80' von Rodnay Zaks und

das neue TCS Colour-Genie-ROM-Listing mit 360 Seiten Umfang!

Regierungsspiel

Auf den folgenden Seiten finden Sie das Listing einer sehr interessanten Regierungssimulation.
Bis zu sechs Spieler konnen dort um die Wette spielen.
Man kann auch alleine spielen, jedoch ist dies naturlich nicht so spannend.
Wenn Sie auch einen Sinn fur anspruchsvolle Spiele haben, werden Sie an diesem Programm lange Zeit Freude haben.

Achtung: Das Regierungsspiel benotigt 32K RAM.

Das Programm gibt auf Wunsch eine kurze Einführung (s. Zeile 7330 ff), jedoch sollen hier noch einige zusatzliche Bemerkungen zum Spielverlauf gemacht werden.

Das Ziel jedes Spielers ist, als erster Konig bzw. Konigin zu werden. Im Verlauf des Spiels wird standig die Leistung jedes Spielers ausgewertet und ihm danach ein Titel zugewiesen.
Die Titel sind:
Fur einen Mann: SIR, BARON, GRAF, MARQUIS, HERZOG, GROSSHERZOG PRINZ und *KONIG*
Fur eine Frau. LADY, BARONESS, GRAFIN, MARQUISE, HERZOGIN, GROSSHERZOGIN, PRINZESSIN und *KONIGIN*
Einen Titel den man einmal erreicht hat, kann man nicht mehr verlieren.
Wie im richtigen Leben, hat man auch hier nicht beliebig viel Zeit um Konig bzw. Konigin zu werden. (Es sei denn man andert Zeile 560.)

Zu Beginn geben Sie die Zahl der Spieler, deren Namen und die Spielstarke ein.
Jeder Spieler spielt dann eine Runde (ein Jahr).
Das 1. Bild zeigt Getreidevorrate, Getreidepreis, Landpreis u.s.w. an. Hier kann man Getreide und Land kaufen oder verkaufen. Interessant ist die Spekulation mit Getreide und Land uber ein Jahr hinaus. Achten Sie also auf die Preise !
Anschließend geben Sie Getreide an die Untertanen, hier sollten Sie nicht weniger als die angezeigte Mindestmenge ausgeben.
Beachten Sie, daß man mindestens 20 Prozent seines Getreidevorrats zuruckhalten muß, und daß man aber auch mindestens 20 Prozent seines Getreidevorrats ausgeben muß.
Wenn man wesentlich mehr als die gefordete Menge Getreide an die Bevolkerung ausgibt, so ziehen weitere Burger zu. Dies macht sich positiv bei den Staatseinkunften bemerkbar. Nach dem Bildschirm, der zeigt, wieviele Untertanen geboren wurden etc., folgt der Bildschirm, der die gewahlten Steuersatze und die Harte der Gerichtsbarkeit anzeigt.

Zur Gerichtsbarkeit ist zu sagen, daß großere Harte mehr Geld einbringt, jedoch das wirtschaftliche Wachstum dadurch bremmst, daß jedes Jahr Untertanen dem Gericht entfliehen. Außerdem wirkt sich harte Justiz ungunstig auf den Titel aus.

Anschließend zeichnet das Programm eine Landkarte.
Die Große der Karte zeigt Ihnen die Große Ihres Landes.
Oben links sehen Sie eine Turm. Dieser Turm zeigt die Starke Ihrer Verteidigung - schrumpft er, sollten Sie sofort Soldaten kaufen, sonst kann es passieren, daß die anderen Spieler Ihnen Land abnehmen.
Am rechten Rand der Karte wird ein stilisiertes Pferd mit einem Landarbeiter gezeichnet. Wenn dieser an der oberen Grenze steht, wird das gesamte Land genutzt. Ist dies nicht der Fall, empfiehlt es sich durch hohere Getreideasugaben mehr Untertanen zu bekommen.
Wenn man Palast, Kathedrale, Marktplatze oder Wollfabriken gekauft hat, werden diese ebenfalls in die Karte eingezeichnet.

Nach der Karte folgt eine Liste von moglichen Staatsanschaffungen. Drucken Sie die jeweilige Zahlentaste, um das gewunschte Gut zu kaufen. Mit <6> kann man hier auch eine Vergleichsubersicht bekommen.

Sie konnen Ihre Staatskasse auch in gewissem Maße verschulden, jedoch mussen Sie fur Ihre Schulden Zinsen zahlen.
Ist die Verschuldung zu hoch droht Ihnen der Bankrott !

Das Programmlisting:

```
10 RANDOM
20 CLEAR 700
30 DEFINT A-J, M-Q, T, V, W
40 Y(0)=1400
50 DATA"Sir ","Baron ","Graf ","Marquis ","Herzog ","Grossherzog ","Pr
inz ","* Koenig *"
60 DATA"Lady ","Baroness ","Graefin ","Marquise ","Herzogin ","Grosshe
rzogin ","Prinzessin ","* Koenigin *"
70 DATA"Santa Paravia","Fiumaccio","Torricella","Molinetto","Fontanile
","Romagna"
80 CLS
90 PRINT
100 COLOUR 16
110 PRINT § 326, CHR$(23);"Santa Paravia und Fiumaccio"
120 COLOUR 5
```

```
130 PRINT § 366, STRING$(27, 217)
140 FOR A=0 TO 555
150 NEXT
160 CLS
170 PRINT
180 FOR A=1 TO 16
190 READ A$
200 NEXT
210 COLOUR 16
220 PRINT"Wieviele Mitspieler ?"
230 COLOUR 5
240 PRINT"(Eingabe zwischen 1 und 6)";
250 GOSUB 7290
260 F=VAL(A$)
270 IF F<1 OR F>6 THEN CLS: GOTO 210
280 FOR A=1 TO F
290 READ T$(A)
300 CLS
310 PRINT
320 PRINT"Wer regiert ";T$(A);
330 COLOUR 16
340 INPUT N$(A)
350 COLOUR 5
360 N$(A)=N$(A)+" von " +T$(A)
370 CLS
380 PRINT"Ist ";N$(A)
390 PRINT"ein ";
400 COLOUR 16
410 PRINT"Mann ";
420 COLOUR 5
430 PRINT"oder eine ";
440 COLOUR 16
450 PRINT"Frau ";
460 COLOUR 5
470 PRINT"?";
480 V(A)=0
490 GOSUB 7290
500 IF A$<>"M" AND A$<>"F" AND A$<>"m" AND A$ <>"f" THEN 480
510 IF LEFT$(A$, 1)="F" OR LEFT$(A$, 1)="f" THEN V(A)=8
520 G(A)=25
530 H(A)=10
540 I(A)=5
550 J(A)=2
560 O(A)=1420+RND(35)
570 K(A)=1000
580 L(A)=10000
590 R(A)=5000
600 T(A)=1
610 U(A)=1
620 N(A)=4
630 P(A)=25
640 Q(A)=5
```

```
650 M(A)=25
660 S(A)=2000
670 NEXT
680 FOR A=1 TO F
690 RESTORE
700 B=V(A)+T(A)
710 FOR C=1 TO B
720 READ T$(A)
730 NEXT C
740 NEXT A
750 PRINT
760 PRINT
770 PRINT"Wuenschen Sie eine Einweisung ? (J/N)";
780 GOSUB 7290
790 IF A$="N" OR A$="n" THEN 820
800 IF A$<>"J" AND A$<>"j" THEN 750: GOSUB 7330
810 GOSUB 7330
820 CLS
830 COLOUR 16
840 PRINT"1:Anfaenger,        2:Fortgeschrittener,  3:Profi,            4:
Grossmeister"
850 COLOUR 5
860 PRINT
870 PRINT"Eingabe der Spielstaerke";
880 GOSUB 7290
890 U(0)=VAL(A$)
900 IF U(0)<1U(0)=1
910 IF U(0)>4U(0)=4
920 U(0)=U(0)+5
930 E=E+1
940 IF T(E)=-1 THEN E=E+1
950 IF(T(1)<1)AND(T(2)<1)AND(T(3)<1)AND(T(4)<1)AND(T(5)<1)AND(T(6)<1)T
HEN 7660
960 IF E>F THEN E=0: Y(0)=Y(0)+1: GOTO 930
970 IF Y(0)>O(E)THEN 930
980 IF Y(0)=O(E)THEN 1340
990 GOSUB 1780
1000 GOSUB 2860
1010 GOSUB 3720
1020 GOSUB 3300
1030 GOSUB 4540
1040 GOSUB 5930
1050 GOSUB 6410
1060 GOTO 930
1070 CLS
1080 PRINT"Adelige Soldaten Klerus Haendler"
1090 PRINT
1100 FOR A=1 TO F
1110 COLOUR 5
1120 PRINT T$(A); N$(A)
1130 COLOUR 16
1140 COLOUR 2
1150 PRINT N(A); TAB(7)P(A); TAB(16)Q(A); TAB(23)M(A)
```

```
1160 COLOUR 16
1170 NEXT
1190 COLOUR 5
1200 PRINT STRING$(40, 217);
1210 COLOUR 16
1230 PRINT"Untertanen  Land    Staatskasse"
1240 FOR A=1 TO F
1250 COLOUR 5
1260 PRINT T$(A); N$(A)
1270 COLOUR 2
1280 PRINT S(A); TAB(11)L(A); TAB(21)K(A)
1290 NEXT A
1300 COLOUR 16
1310 PRINT
1320 INPUT"  (Druecke RETURN)";A$
1330 RETURN
1340 CLS
1350 PRINT
1360 PRINT"Sehr schlechte Nachrichten"
1370 PRINT
1380 PRINT T$(E); N$(E)
1390 PRINT"ist gerade gestorben,"
1400 T(E)=-1
1410 Y=RND(8)
1420 IF Y(0)>1450 PRINT"nach langer Regentschaft und in hohem   Alter"
: GOTO 1480
1430 IF Y<4 PRINT"an Lungenentzuendung nach einem kalten  Winter in se
inem Schloss."
1440 IF Y=5 PRINT"waehrend einer Pockenepidemie."
1450 IF Y=4 PRINT"an Thyphus nach Genuss vom schmutzigem  Wasser."
1460 IF Y=6 PRINT"nach einem Raubueberfall waehrend       einer Reise.
"
1470 IF Y>6 PRINT"an Fleischvergiftung."
1480 PRINT
1490 INPUT"(Druecke RETURN)";A$
1500 IF F=1 THEN 7650
1510 GOSUB 4540
1520 GOSUB 1070
1530 GOTO 930
1540 I!=INT(I!)
1550 RETURN
1560 C!=INT(C!)
1570 RETURN
1580 S!=INT(S!)
1590 RETURN
1600 K(E)=INT(K(E))
1610 RETURN
1620 Z=RND(A)*S(E)/100
1630 Z%=Z
1640 COLOUR 2
1650 PRINT Z%;
1660 COLOUR 16
1670 PRINT"Sklaven in diesem Jahr geboren"
1680 S(E)=S(E)+Z%
```

```
1690 RETURN
1700 Z=RND(A)*S(E)/100
1710 Z%=Z
1720 COLOUR 3
1730 PRINT Z%;
1740 COLOUR 16
1750 PRINT"Sklaven in diesem Jahr gestorben."
1760 S(E)=S(E)-Z%
1770 RETURN
1780 W=( RND(5)+RND(6))/2
1790 ON W GOTO 1800, 1820, 1840, 1860, 1880
1800 W$="Trocken, eine Hungersnot droht !!"
1810 GOTO 1900
1820 W$="Schlechtes Wetter, duerftige Ernte"
1830 GOTO 1900
1840 W$="Normales Wetter, durchschnittliche Ernte"
1850 GOTO 1900
1860 W$="Schoenes Wetter, gute Ernte"
1870 GOTO 1900
1880 W$="Exzellentes Wetter, hervorragende Ernte"
1890 GOTO 1900
1900 R=RND(50)
1910 R(E)=(R(E)*100-R(E)*R)/100
1920 X=L(E)
1930 Y=(S(E)-D(E)*100)*5
1940 IF Y<0Y=0
1950 IF Y<X THEN X=Y
1960 Y=R(E)*2
1970 IF Y<X THEN X=Y
1980 R(E)=R(E)-X/2
1990 Y=W-.5
2000 H!=X*Y
2010 R(E)=R(E)+H!
2020 D!=N(E)*100+C(E)*40+M(E)*30+P(E)*10+S(E)*5
2030 L=(3*W+RND(6)+RND(6)+10)/10
2040 IF W=1 THEN L=L-1
2050 IF H!<1Y=2: GOTO 2080
2060 Y=D!/H!
2070 IF Y>2Y=2
2080 IF Y<.8Y=.8
2090 L=L*Y
2100 L=INT(L*10)
2110 L=L/10
2120 Z=6-W
2130 G=(Z*3+RND(5)+RND(5))/5*Y*20
2140 RETURN
2150 COLOUR 5
2160 PRINT STRING$(40, 217);
2170 COLOUR 16
2180 PRINT"Ratten vernichteten ";
2190 COLOUR 3
2200 PRINT R;
2210 COLOUR 16
2220 PRINT" % deiner"
2230 PRINT"Getreidereserven"
```

```
2240 PRINT
2250 PRINT W$
2260 PRINT"          (";
2270 COLOUR 2
2280 PRINT H!;
2290 COLOUR 16
2300 PRINT" Zentner )"
2310 COLOUR 5
2320 PRINT STRING$(40, 217);
2330 IF K(E)<32766 GOSUB 1600
2340 COLOUR 16
2350 PRINT"Getreide  Getreide  Getreide-  Land-"
2360 PRINT"Vorraete  Bedarf    preis      preis"
2370 COLOUR 2
2380 PRINT R(E); TAB(10)D!; TAB(20)G; TAB(31)L
2390 COLOUR 16
2400 PRINT"Zentner   Zentner   1000 Z.    Hektar"
2410 COLOUR 5
2420 PRINT STRING$(40, 217);
2430 COLOUR 16
2440 PRINT"Staatskasse: ";
2450 IF K(E)>0 THEN COLOUR 2 ELSE COLOUR 3
2460 PRINT K(E)
2470 RETURN
2480 J=(J(E)*300-500)*T(E)
2490 ON J(E)GOTO 2500, 2520, 2540, 2560
2500 J$="sehr gerecht"
2510 GOTO 2570
2520 J$="gemaessigt"
2530 GOTO 2570
2540 J$="streng"
2550 GOTO 2570
2560 J$="sehr streng"
2570 Y=150-G(E)-H(E)-I(E)
2580 IF Y<1Y=1
2590 C!=(N(E)*180+Q(E)*75+M(E)*20)*(Y/100)+U(E)*100
2600 S!=(N(E)*50+M(E)*25+U(E)*10)*(Y/100)*(5-J(E))/2
2610 I!=N(E)*250+U(E)*20+(10*J(E)*N(E))*(Y/100)
2620 C!=C!*G(E)/100
2630 IF C!<32760 GOSUB 1560
2640 S!=S!*H(E)/100
2650 IF S!<32760 GOSUB 1580
2660 I!=I!*I(E)/100
2670 IF I!<32760 GOSUB 1540
2680 COLOUR 16
2690 PRINT"Staatseinnahmen     ";
2700 COLOUR 2
2710 PRINT J+C!+S!+I!;
2720 COLOUR 16
2730 PRINT"Gulden"
2740 COLOUR 5
2750 PRINT STRING$(40, 217);
2760 COLOUR 16
2770 PRINT"Zollge-   Umsatz-   Eink.  Gericht"
```

```
2780 PRINT"buehren   steuer    steuer"
2790 COLOUR 2
2800 PRINT G(E);"%",H(E);"%",I(E);"%":TAB(27)J$
2810 PRINT C!, S!, I!, J;"Gl."
2820 COLOUR 5
2830 PRINT STRING$(40, 217);
2840 COLOUR 16
2850 RETURN
2860 CLS
2870 PRINT
2880 COLOUR 16
2890 PRINT T$(E); N$(E)
2900 COLOUR 5
2910 GOSUB 2150
2920 COLOUR 5
2930 PRINT STRING$(40, 217);
2940 COLOUR 16
2950 PRINT"1:Getreide kaufen","2:Getreide verkaufen";
2960 PRINT"3:Land kaufen     ","4:Land verkaufen"
2970 PRINT"Jede andere Taste zum Fortfahren"
2980 COLOUR 5
2990 PRINT STRING$(40, 217);
3000 COLOUR 16
3010 GOSUB 7290
3020 I!=VAL(A$)
3030 PRINT
3040 IF I!>4 THEN CLS: GOTO 2950
3050 IF I!<1 THEN RETURN
3060 ON I!GOTO 3070, 3160, 3210, 3250
3070 PRINT
3080 INPUT"Wieviel Getreide kaufen";I!
3090 K(E)=K(E)-(I!*G/1000)
3100 R(E)=R(E)+I!
3110 CLS
3120 PRINT
3130 PRINT T$(E); N$(E)
3140 GOSUB 2320
3150 GOTO 2920
3160 INPUT"Wieviel Getreide verkaufen";I!
3170 IF I!>R(E)PRINT"... so viel hast Du nicht": PRINT: GOTO 3160
3180 K(E)=K(E)+(I!*G/1000)
3190 R(E)=R(E)-I!
3200 GOTO 3110
3210 INPUT"Wieviel Hektar kaufen";I!
3220 L(E)=L(E)+I!
3230 K(E)=K(E)-(I!*L)
3240 GOTO 3110
3250 INPUT"Wieviel Hektar verkaufen";I!
3260 IF I!>(L(E)-5000)PRINT"... so viel kannst Du nicht verkaufen": GO
TO 3250
3270 L(E)=L(E)-I!
3280 K(E)=K(E)+(I!*L)
3290 GOTO 3110
```

```
3300 CLS
3310 PRINT
3320 PRINT T$(E); N$(E)
3330 PRINT
3340 GOSUB 2480
3350 PRINT
3360 PRINT"1. Zollgebuehren"
3370 PRINT"2. Umsatzsteuer"
3380 PRINT"3. Eink.steuer"
3390 PRINT"4. Gericht"
3400 PRINT
3410 PRINT"(Tippe Nummer zum Aendern"
3420 PRINT" 0 zur Fortsetzung)";
3430 GOSUB 7290
3440 I=VAL(A$)
3450 PRINT
3460 IF I>4 CLS: GOTO 3350
3470 IF I<1 THEN 3680
3480 ON I GOTO 3490, 3540, 3580, 3620
3490 INPUT"Neue Zollgebuehren (0 bis 100 %)";I
3500 IF I>100I=100
3510 IF I<0I=0
3520 G(E)=I
3530 GOTO 3300
3540 INPUT"Neue Umsatzsteuer (0 bis 50 %)";I
3550 IF(I>50)OR(I<0)I=5
3560 H(E)=I
3570 GOTO 3300
3580 INPUT"Neue Einkommensteuer (0 bis 25 %)";I
3590 IF(I<0)OR(I>25)THEN I=0
3600 I(E)=I
3610 GOTO 3300
3620 PRINT"Gericht: 1.sehr gerecht  2.gemaessigt   3.streng  4.sehr st
reng";
3630 GOSUB 7290
3640 I=VAL(A$)
3650 IF(I>4)OR(I<1)THEN I=1
3660 J(E)=I
3670 GOTO 3300
3680 K(E)=K(E)+C!+S!+I!+J
3690 IF K(E)<0 THEN K(E)=K(E)*1.5
3700 IF K(E)<(-10000*T(E)) THEN 6900
3710 RETURN
3720 COLOUR 16
3730 INPUT"Wieviel Getreide gibst du aus";G!
3740 IF G!<(R(E)/5)PRINT"Du musst mindestens 20% ausgeben !": GOTO 373
0
3750 IF G!>(R(E)-(R(E)/5)) PRINT"Du musst mind. 20% zurueckbehalten":
GOTO 3730
3760 R(E)=R(E)-G!
3770 CLS
3780 PRINT
3790 PRINT T$(E); N$(E)
3800 PRINT
```

```
3810 Z=G!/D!-1
3820 IF Z>0Z=Z/2
3830 IF Z>.25Z=Z/10+.25
3840 Z%=50-G(E)-H(E)-I(E)
3850 IF Z%<0Z%=Z%*J(E)
3860 Z%=Z%/10
3870 IF Z%>0Z%=Z%+3-J(E)
3880 Z=Z+(Z%/10)
3890 IF Z>.5Z=.5
3900 IF G!<(D!-1)THEN 4250
3910 A=7
3920 GOSUB 1620
3930 A=3
3940 GOSUB 1700
3950 IF(G(E)+H(E))<35M(E)=M(E)+RND(4)
3960 IF I(E)<RND(20)N(E)=N(E)+RND(2)-1: Q(E)=Q(E)+RND(3)-1
3970 IF G!<(D!+D!*.3)THEN 4160
3980 Z%=S(E)/1000
3990 Z=(G!-D!)/D!*10
4000 Z=Z*Z%*RND(25)+RND(40)
4010 IF Z>32000Z=32000
4020 Z%=Z
4030 Z=RND(Z%)
4040 COLOUR 2
4050 PRINT Z;
4060 COLOUR 16
4070 PRINT"Untertanen kamen in die Stadt."
4080 S(E)=S(E)+Z
4090 U(E)=U(E)+.5
4100 Z%=Z/5
4110 Z=RND(Z%)
4120 IF Z>50Z=50
4130 M(E)=M(E)+Z
4140 N(E)=N(E)+1
4150 Q(E)=Q(E)+2
4160 IF J(E)<3 THEN 4240
4170 J!=S(E)/100*(J(E)-2)*(J(E)-2)
4180 J!=RND(J!)
4190 S(E)=S(E)-J!
4200 COLOUR 3
4210 PRINT J!;
4220 COLOUR 16
4230 PRINT"Untertanen entfliehen dem Gericht."
4240 GOTO 4350
4250 X=(D!-G!)/D!*100-9
4260 X%=X
4270 IF X>65 THEN X=65: M(E)=M(E)/2
4280 IF X<0X%=0: X=0
4290 A=3
4300 GOSUB 1620
4310 A=X%+8
4320 GOSUB 1700
4330 IF Z%>1000 THEN U(E)=U(E)/2
```

```
4340 GOTO 4160
4350 Z=A(E)*75
4360 K(E)=K(E)+Z
4370 IF Z>0 PRINT"Dein Markt brachte";: COLOUR 2: PRINT Z;: COLOUR 16:
 PRINT"Gulden Pacht ein"
4380 IF S(E)<32766S!=S(E): GOSUB 1580: S(E)=S!
4390 Z=D(E)*(55+RND(250))
4400 IF Z>0K(E)=K(E)+Z: PRINT"Deine Wollfabrik hatte";: COLOUR 2: PRIN
T Z;: COLOUR 16: PRINT"Gulden Gewinn zu verzeichnen"
4410 Z=P(E)*3
4420 PRINT
4430 PRINT"Du hast Deinen Soldaten";Z;"Gulden Sold"
4440 PRINT"ausbezahlt."
4450 K(E)=K(E)-Z
4460 IF(L(E)/1000)>P(E)THEN 7070
4470 IF(L(E)/500)<P(E)THEN 4520
4480 FOR A=1 TO F
4490 IF A=E THEN 4510
4500 IF P(A)>(P(E)*2.4)THEN 7070
4510 NEXT
4520 INPUT"(Druecke RETURN)";A$
4530 RETURN
4540 CLS
4550 COLOUR 16
4560 PRINT"Jahr";
4570 COLOUR 2
4580 PRINT Y(0)
4590 COLOUR 16
4600 PRINT
4610 PRINT T$(E); N$(E)
4620 PRINT § 400,"Nach Karte weiter mit <RETURN>"
4630 PRINT § 160,;
4640 INPUT"<RETURN> druecken fuer Landkarte";XX$
4650 FCLS
4660 FGR
4670 FCOLOUR 3
4680 L%=(L(E)/1000)
4690 IF L%<10X=80: Y=27: GOTO 4770
4700 IF L%<30X=80: Y=27-(L%-10): GOTO 4770
4710 IF L%<50X=60: Y=27-(L%-30): GOTO 4770
4720 IF L%<70X=40: Y=27-(L%-50): GOTO 4770
4730 IF L%<90X=20: Y=27-(L%-70): GOTO 4770
4740 IF L%<110X=1: Y=27-(L%-90): GOTO 4770
4750 X=1
4760 Y=7
4770 REM
4780 PLOT X, Y TO 127, Y
4790 PLOT X, Y TO X, 47
4800 IF(P(E)-5)<(L(E)/1000)THEN 5060
4810 FOR A=X+1 TO X+6
4820 FOR B=Y+1 TO Y+5
4830 PLOT A, B
4840 NEXT B
4850 NEXT A
```

```
4860 PLOT X, Y-1
4870 PLOT X+2, Y-1
4880 PLOT X+4, Y-1
4890 PLOT X+6, Y-1
4900 IF(P(E)/2)<(L(E)/1000)THEN 5060
4910 FOR A=X+7 TO X+10
4920 FOR B=Y+1 TO Y+5
4930 PLOT A, B
4940 NEXT B
4950 NEXT A
4960 PLOT X+8, Y-1
4970 PLOT X+10, Y-1
4980 NPLOT X+3, Y+2
4990 NPLOT X+7, Y+4
5000 PLOT X+1, Y-1
5010 PLOT X+9, Y-1
5020 PLOT X, Y-2
5030 PLOT X+2, Y-2
5040 PLOT X+8, Y-2
5050 PLOT X+10, Y-2
5060 Z=C(E)+1
5070 IF Z>7 THEN Z=7
5080 ON Z GOTO 5410, 5340, 5270, 5230, 5180, 5130, 5090
5090 REM
5100 PLOT 96, 30 TO 110, 30
5110 NPLOT 102, 30
5120 NPLOT 104, 30
5130 FOR A=96 TO 99
5140 FOR B=24 TO 29
5150 PLOT A, B
5160 NEXT B
5170 NEXT A
5180 FOR A=107 TO 110
5190 FOR B=24 TO 29
5200 PLOT A, B
5210 NEXT B
5220 NEXT A
5230 REM
5240 PLOT 103, 22 TO 103, 24
5250 PLOT 102, 23
5260 PLOT 104, 23
5270 FOR A=101 TO 105
5280 FOR B=25 TO 26
5290 PLOT A, B
5300 NEXT B
5310 NEXT A
5320 NPLOT 101, 25
5330 NPLOT 105, 25
5340 FOR A=100 TO 106
5350 FOR B=27 TO 29
5360 PLOT A, B
5370 NEXT B
5380 NEXT A
```

```
5390 NPLOT 102, 29
5400 NPLOT 104, 29
5410 Z=B(E)*2
5420 IF Z=0 THEN 5540
5430 IF Z>10 THEN PLOT 80, 33: PLOT 82, 33: PLOT 96, 33: PLOT 98, 33:
PLOT 81, 32: PLOT 97, 32
5440 IF Z>8 THEN Z=9: FOR A=87 TO 91: FOR B=31 TO 33: PLOT A, B: NEXT
B, A: NPLOT 88, 33: NPLOT 90, 32: PLOT 88, 30: PLOT 90, 30: PLOT 89, 2
9
5450 FOR A=(89-Z)TO(89+Z)
5460 FOR B=34 TO 36
5470 PLOT A, B
5480 NEXT B
5490 NEXT A
5500 FOR A=(90-Z)TO(90+Z)STEP 2
5510 NPLOT A, 35
5520 NEXT A
5530 NPLOT 89, 36
5540 Z=S(E)-D(E)*100
5550 IF Z<1 THEN Z=1
5560 Z=Z*5/L(E)*10+1
5570 IF Z>10 THEN Z=10
5580 Z=(Z/10)*(45-Y)
5590 Z=INT(47-Z)
5600 PLOT 119, Z TO 127, Z
5610 NPLOT 122, Z
5620 NPLOT 123, Z
5630 NPLOT 125, Z
5640 FOR A=119 TO 127 STEP 2
5650 PLOT A, Z+1
5660 NEXT A
5670 PLOT 118, Z-1
5680 PLOT 127, Z-1
5690 Z=A(E)*2
5700 IF Z=0 THEN 5780
5710 IF Z>((126-X)-2)Z=((126-X)-2)
5720 FOR A=X TO X+Z STEP 2
5730 PLOT A, 39
5740 PLOT A+1, 39
5750 PLOT A+1, 40
5760 PLOT A+1, 41
5770 NEXT A
5780 Z=D(E)
5790 IF Z=0 THEN 5890
5800 IF Z>(126-X)Z=126-X
5810 FOR A=126-Z TO 127
5820 FOR B=45 TO 47
5830 PLOT A, B
5840 NEXT B
5850 NEXT A
5860 FOR A=127-Z TO 126 STEP 2
5870 NPLOT A, 46
5880 NEXT A
```

```
5890 REM
5900 GOSUB 7290
5910 LGR
5920 RETURN
5930 CLS
5940 PRINT
5950 PRINT T$(E); N$(E)
5960 PRINT"Staatsanschaffungen"
5970 PRINT
5980 PRINT"1. Marktplatz                     1000 Gulden"
5990 PRINT"2. Wollfabrik                     2000 Gulden"
6000 PRINT"3. Palast (Teil)                  3000 Gulden"
6010 PRINT"4. Kathedrale  (Teil)             5000 Gulden"
6020 PRINT"5. 20 Soldaten                     500 Gulden"
6030 PRINT
6040 PRINT"Du hast ";
6050 IF K(E)<0 THEN COLOUR 3: PRINT K(E); ELSE COLOUR 2: PRINT K(E);
6060 COLOUR 16
6070 PRINT"Gulden"
6080 PRINT
6090 PRINT"0 zur Fortsetzung, 6 fuer Vergleich"
6100 PRINT
6110 PRINT"Deine Wahl ?";
6120 GOSUB 7290
6130 I=VAL(A$)
6140 CLS
6150 IF I<1 RETURN
6160 IF I>5 GOSUB 1070: GOTO 5930
6170 ON I GOTO 6220, 6180, 6270, 6320, 6370
6180 D(E)=D(E)+1
6190 K(E)=K(E)-2000
6200 U(E)=U(E)+.25
6210 GOTO 5930
6220 A(E)=A(E)+1
6230 M(E)=M(E)+5
6240 K(E)=K(E)-1000
6250 U(E)=U(E)+.1
6260 GOTO 5930
6270 B(E)=B(E)+1
6280 N(E)=N(E)+RND(2)
6290 K(E)=K(E)-3000
6300 U(E)=U(E)+.5
6310 GOTO 5930
6320 C(E)=C(E)+1
6330 Q(E)=Q(E)+RND(6)
6340 K(E)=K(E)-5000
6350 U(E)=U(E)+1
6360 GOTO 5930
6370 P(E)=P(E)+20
6380 S(E)=S(E)-20
6390 K(E)=K(E)-500
6400 GOTO 5930
6410 Z=0
```

```
6420 A=A(E)
6430 GOSUB 6780
6440 A=B(E)
6450 GOSUB 6780
6460 A=C(E)
6470 GOSUB 6780
6480 A=O(E)
6490 GOSUB 6780
6500 A=K(E)/5000
6510 GOSUB 6780
6520 A=L(E)/6000
6530 GOSUB 6780
6540 A=M(E)/50
6550 GOSUB 6780
6560 A=N(E)/5
6570 GOSUB 6780
6580 A=P(E)/50
6590 GOSUB 6780
6600 A=Q(E)/10
6610 GOSUB 6780
6620 A=S(E)/2000
6630 GOSUB 6780
6640 A=U(E)/5
6650 GOSUB 6780
6660 A=Z/U(O)-J(E)+1
6670 A=INT(A)
6680 IF A>8A=8
6690 IF(Y(O)+2)=O(E)T(E)=T(E)+1
6700 IF T(E)>=A THEN 6770
6710 T(E)=A
6720 RESTORE
6730 FOR B=1 TO(T(E)+V(E))
6740 READ T$(E)
6750 NEXT
6760 IF T(E)=8 THEN 6820
6770 RETURN
6780 IF A>10: A=10
6790 A=INT(A)
6800 Z=Z+A
6810 RETURN
6820 CLS
6830 PRINT
6840 PRINT"Das Spiel ist vorbei:"
6850 PRINT T$(E); N$(E)
6860 PRINT"hat gewonnen"
6870 GOSUB 4680
6880 GOSUB 1070
6890 GOTO 7670
6900 CLS
6910 PRINT
6920 PRINT T$(E); N$(E);" ist bankrott"
6930 PRINT
6940 PRINT"Glaeubiger haben das meiste Deines      Eigentums beschlagn
```

```
ahmt !"
6950 PRINT
6960 INPUT"(Druecke RETURN)";A$
6970 A(E)=0
6980 B(E)=0
6990 C(E)=0
7000 D(E)=0
7010 L(E)=6000
7020 U(E)=1
7030 K(E)=100
7040 M(E)=M(E)/2
7050 R(E)=4000
7060 RETURN
7070 Z=0
7080 FOR A=1 TO F
7090 IF A=E THEN 7130
7100 IF P(A)<P(E)THEN 7130
7110 IF P(A)<(1.2*(L(A)/1000)) THEN 7130
7120 IF P(A)>P(Z)Z=A
7130 NEXT
7140 IF Z=0T$(0)=" Baron ": N$(0)="Peppone von Monterana hat": A!=RND(
9000)+1000: GOTO 7160
7150 A!=P(Z)*1000-L(Z)/3
7160 IF A!>(L(E)-5000)A!=(L(E)-5000)/2
7170 PRINT T$(Z); N$(Z)
7180 PRINT"angegriffen und beschlagnahmt";A!
7190 PRINT"Hektar Land!"
7200 L(Z)=L(Z)+A!
7210 L(E)=L(E)-A!
7220 Z=RND(40)
7230 IF Z>(P(E)-15)Z=P(E)-15
7240 PRINT T$(E); N$(E); Z
7250 PRINT"Soldaten in der Schlacht gefallen."
7260 P(E)=P(E)-Z
7270 INPUT"(Druecke RETURN)";A$
7280 RETURN
7290 A$=INKEY$
7300 A$=INKEY$
7310 IF A$="" THEN 7300
7320 RETURN
7330 CLS
7340 COLOUR 5
7350 PRINT"Santa Paravia und Fuimaccio"
7360 PRINT STRING$(40, 217);
7370 COLOUR 16
7380 PRINT"Du regierst einen italienischen Stadt-"
7390 PRINT"staat aus dem 15. Jahrhundert. Regierst"
7400 PRINT"du gut, bekommst du hoehere Titel. Der"
7410 PRINT"1. Spieler, der Koenig(in) wird, hat"
7420 PRINT"gewonnen. Regierst du schlecht, wird"
7430 PRINT"deine Lebenszeit nicht reichen, um "
7440 PRINT"Koenig(in) zu werden. Der Computer"
7450 PRINT"zeichnet eine Karte deines Reiches."
```

```
7460 PRINT"Die Groesse des Turmes links oben zeigt"
7470 PRINT"die Verteidigungskraft an. Wird der"
7480 PRINT"Turm kleiner, brauchst du mehr Soldaten"
7490 PRINT"Wenn das Pferd mit dem Landarbeiter"
7500 PRINT"(oben rechts) die obere Grenze erreicht,";
7510 PRINT"wird das ganze Land genutzt. Wenn du"
7520 PRINT"zuwenig Getreide ausgibst, sterben"
7530 PRINT"deine Untertanen. Gibst du viel"
7540 PRINT"Getreide aus, kommen neue Einwohner"
7550 PRINT"hinzu. Hohe Steuern bringen mehr Geld"
7560 PRINT"ein, verlangsamen aber das wirtschaft-"
7570 PRINT"liche Wachstum."
7580 COLOUR 5
7590 PRINT STRING$(40, 217);
7600 PRINT"(Druecke <RETURN>, um zu beginnen)";
7610 COLOUR 16
7620 INPUT A$
7630 CLS
7640 RETURN
7650 GOSUB 4540
7660 GOSUB 1070
7670 PRINT
7680 PRINT"SPIELENDE. <RETURN> fuer neues Spiel ";
7690 INPUT A$
7700 GOTO 10
```

Ein Super-Spielprogramm in Maschinensprache: BALLON

Bei diesem Spiel geht es darum, zwei Mannchen mit einem Trampolin so zu bewegen, daß sie ein drittes Mannchen damit hochschleudern. Dabei soll es moglichst viele der Ballons zum zerplatzen bringen.

Es gewinnt dabei langsam immer mehr an Hohe und wird immer schneller. Fur jeden zerplatzten Luftballon gibt es 50 Punkte. Die Steuerung der beiden unteren Mannchen geschieht mit den Pfeiltasten nach links bzw. nach rechts.

Noch ein Tip zum spielen:
Versuchen Sie das Mannchen moglichst waagerecht fliegen zu lassen, da Sie so leichter die Luftballons holen konnen. Dazu lassen Sie das Mannchen am linken bzw. rechten Rand des Trampolins aufkommen.

Das Spiel liegt im Speicherbereich zwischen 4800H und 57FFH, also im Grafikspeicher (LGR-Modus) des Colour-Genie.
Der erste Bereich von 4800H bis 4BFFH enthalt die Grafik fur das Spiel. Danach folgen Routinen zur Bewegung der Ballons und Mannchen. Ab 5100H ist der Schriftzug BALLON abgelegt.
Er wird bei der Anfangsgrafik in das normale Bild kopiert.
Ab 5200H bis 55FFH ist die Tabelle fur die definierbaren Zeichen angelegt, die im Anschluß an das Hexlisting noch einmal abgedruckt sind.
Bei 5600H liegt die Haupt-Programmschleife mit der Tastaturabfrage fur die Bewegung. Die Startadresse ist 5790H. Dort wird jedoch nur der High-Score mit Null belegt und sodann zur Anfangsgrafik verzweigt.

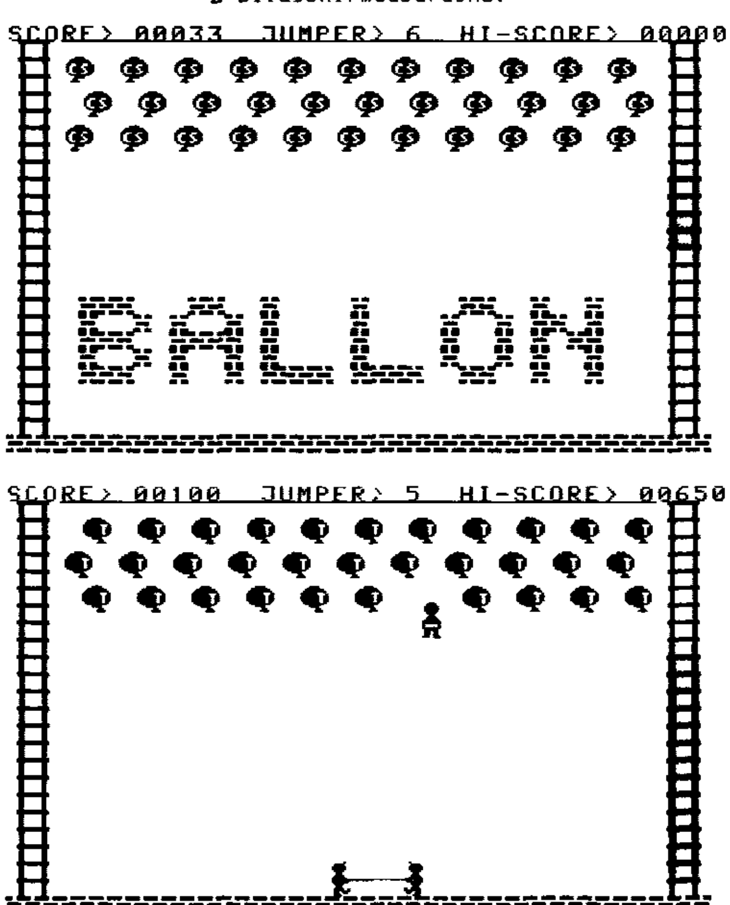
Zur besseren Orientierung im Programm, hier die als Variablen benutzten Speicherzellen und ihre Bedeutung:

4288H	- Position der beiden unteren Mannchen
4289H	- Flag fur deren Bewegung
428AH/428BH	- X-Position des Jumpers mit LSB/MSB
428CH/428DH	- Y-Position des Jumpers
428EH	- Zahler fur die Bewegung des Jumpers
4290H/4291H	- X-Richtung des Jumpers mit LSB/MSB
4292H/4293H	- Y-Richtung des Jumpers
4294H/4295H	- Buffer fur X-Richtung
4296H/4297H	- Buffer fur Y-Richtung
4298H	- Zahler fur restliche Ballons (31...0)
429AH/429BH	- Score mit LSB/MSB
429CH/429DH	- High-Score mit LSB/MSB

Auf der nachsten Seite finden Sie 2 Bildschirmausdrucke.
Es folgt dann das Hexlisting, das Sie mit dem Maschinensprachemonitor in Basic (aus Colour-Genie Buch I/II), oder mit dem neuen Maschinensprachemonitor aus diesem Buch, eingeben konnen.

Ballon * Ballon * Ballon * Ballon * Ballon * Ballon * Ballon *
Ballon * Ballon * Ballon * Ballon * Ballon * Ballon * Ballon *
Ballon * Ballon * Ballon * Ballon * Ballon * Ballon * Ballon *
Ballon * Ballon * Ballon * Ballon * Ballon * Ballon * Ballon *
Ballon * Ballon * Ballon * Ballon * Ballon * Ballon * Ballon *
Ballon * Ballon * Ballon * Ballon * Ballon * Ballon * Ballon *

2 Bildschirmausdrucke:



Ballon * Ballon * Ballon * Ballon * Ballon * Ballon * Ballon *
Ballon * Ballon * Ballon * Ballon * Ballon * Ballon * Ballon *
Ballon * Ballon * Ballon * Ballon * Ballon * Ballon * Ballon *
Ballon * Ballon * Ballon * Ballon * Ballon * Ballon * Ballon *
Ballon * Ballon * Ballon * Ballon * Ballon * Ballon * Ballon *
Ballon * Ballon * Ballon * Ballon * Ballon * Ballon * Ballon *

Hexlisting Ballons:

4800	53	43	4F	52	45	3E	20	30
4808	30	30	30	30	20	20	4A	55
4810	4D	50	45	52	3E	20	36	20
4818	20	48	49	2D	53	43	4F	52
4820	45	3E	20	30	30	30	30	30
4828	8C	8D	8E	B7	B7	B7	B7	B7
4830	B7	B7	B7	B7	B7	B7	B7	B7
4838	B7	B7	B7	B7	B7	B7	B7	B7
4840	B7	B7	B7	B7	B7	B7	B7	B7
4848	B7	B7	B7	B7	8C	8D	8E	20
4850	8C	8D	8E	84	85	C0	84	85
4858	C0	84	85	C0	84	85	C0	84
4860	85	C0	84	85	C0	84	85	C0
4868	84	85	C0	84	85	C0	84	85
4870	C0	84	85	C0	8C	8D	8E	20
4878	8C	8D	8E	94	95	C0	94	95
4880	C0	94	95	C0	94	95	C0	94
4888	95	C0	94	95	C0	94	95	C0
4890	94	95	C0	94	95	C0	94	95
4898	C0	94	95	C0	8C	8D	8E	20
48A0	8C	8D	8E	C0	84	85	C0	84
48A8	85	C0	84	85	C0	84	85	C0
48B0	84	85	C0	84	85	C0	84	85
48B8	C0	84	85	C0	84	85	C0	84
48C0	85	C0	84	85	8C	8D	8E	20
48C8	8C	8D	8E	C0	94	95	C0	94
48D0	95	C0	94	95	C0	94	95	C0
48D8	94	95	C0	94	95	C0	94	95
48E0	C0	94	95	C0	94	95	C0	94
48E8	95	C0	94	95	8C	8D	8E	20
48F0	8C	8D	8E	84	85	C0	84	85
48F8	C0	84	85	C0	84	85	C0	84
4900	85	C0	84	85	C0	84	85	C0
4908	84	85	C0	84	85	C0	84	85
4910	C0	84	85	C0	8C	8D	8E	20
4918	8C	8D	8E	94	95	C0	94	95
4920	C0	94	95	C0	94	95	C0	94
4928	95	C0	94	95	C0	94	95	C0
4930	94	95	C0	94	95	C0	94	95
4938	C0	94	95	C0	8C	8D	8E	20
4940	8C	8D	8E	20	20	20	20	20
4948	20	20	20	20	20	20	20	20
4950	20	20	20	20	20	20	20	20
4958	20	20	20	20	20	20	20	20
4960	20	20	20	20	8C	8D	8E	20
4968	8C	8D	8E	20	20	20	20	20
4970	20	20	20	20	20	20	20	20
4978	20	20	20	20	20	20	20	20
4980	20	20	20	20	20	20	20	20
4988	20	20	20	20	8C	8D	8E	20

4990	8C	8D	8E	20	20	20	20	20
4998	20	20	20	20	20	20	20	20
49A0	20	20	20	20	20	20	20	20
49A8	20	20	20	20	20	20	20	20
49B0	20	20	20	20	8C	8D	8E	20
49B8	8C	8D	8E	20	20	20	20	20
49C0	20	20	20	20	20	20	20	20
49C8	20	20	20	20	20	20	20	20
49D0	20	20	20	20	20	20	20	20
49D8	20	20	20	20	8C	8D	8E	20
49E0	8C	8D	8E	20	20	20	20	20
49E8	20	20	20	20	20	20	20	20
49F0	20	20	20	20	20	20	20	20
49F8	20	20	20	20	20	20	20	20
4A00	20	20	20	20	8C	8D	8E	20
4A08	8C	8D	8E	20	20	20	20	20
4A10	20	20	20	20	20	20	20	20
4A18	20	20	20	20	20	20	20	20
4A20	20	20	20	20	20	20	20	20
4A28	20	20	20	20	8C	8D	8E	20
4A30	8C	8D	8E	20	20	20	20	20
4A38	20	20	20	20	20	20	20	20
4A40	20	20	20	20	20	20	20	20
4A48	20	20	20	20	20	20	20	20
4A50	20	20	20	20	8C	8D	8E	20
4A58	8C	8D	8E	20	20	20	20	20
4A60	20	20	20	20	20	20	20	20
4A68	20	20	20	20	20	20	20	20
4A70	20	20	20	20	20	20	20	20
4A78	20	20	20	20	8C	8D	8E	20
4A80	8C	8D	8E	20	20	20	20	20
4A88	20	20	20	20	20	20	20	20
4A90	20	20	20	20	20	20	20	20
4A98	20	20	20	20	20	20	20	20
4AA0	20	20	20	20	8C	8D	8E	20
4AA8	8C	8D	8E	20	20	20	20	20
4AB0	20	20	20	20	20	20	20	20
4AB8	20	20	20	20	20	20	20	20
4AC0	20	20	20	20	20	20	20	20
4AC8	20	20	20	20	8C	8D	8E	20
4AD0	8C	8D	8E	20	20	20	20	20
4AD8	20	20	20	20	20	20	20	20
4AE0	20	20	20	20	20	20	20	20
4AE8	20	20	20	20	20	20	20	20
4AF0	20	20	20	20	8C	8D	8E	20
4AF8	8C	8D	8E	20	20	20	20	20
4B00	20	20	20	20	20	20	20	20
4B08	20	20	20	20	20	20	20	20
4B10	20	20	20	20	20	20	20	20
4B18	20	20	20	20	8C	8D	8E	20
4B20	8C	8D	8E	20	20	20	20	20
4B28	20	20	20	20	20	20	20	20
4B30	20	20	20	20	20	20	20	20

✗4B38	20	20	20	20	20	20	20	20
4B40	20	20	20	20	8C	8D	8E	20
4B48	8C	8D	8E	20	20	20	20	20
4B50	20	20	20	20	20	20	20	20
4B58	20	20	20	20	20	20	20	20
4B60	20	20	20	20	20	20	20	20
4B68	20	20	20	20	8C	8D	8E	20
4B70	8C	8D	8E	20	20	20	20	20
4B78	20	20	20	20	20	20	20	20
4B80	20	20	20	20	20	20	20	20
4B88	20	20	20	20	20	20	20	20
4B90	20	20	20	20	8C	8D	8E	20
4B98	8C	8D	8E	20	20	20	20	20
4BA0	20	20	20	20	20	20	20	20
4BA8	20	20	20	20	20	20	20	20
4BB0	20	20	20	20	20	20	20	20
4BB8	20	20	20	20	8C	8D	8E	20
4BC0	8F	8F	8F	8F	8F	8F	8F	8F
4BC8	8F	8F	8F	8F	8F	8F	8F	8F
4BD0	8F	8F	8F	8F	8F	8F	8F	8F
4BD8	8F	8F	8F	8F	8F	8F	8F	8F
4BE0	8F	8F	8F	8F	8F	8F	8F	20
4BE8	20	20	20	20	20	20	20	20
4BF0	20	20	20	20	20	20	20	20
4BF8	20	20	20	20	20	20	20	20
4C00	21	00	48	11	00	44	01	00
4C08	04	ED	B0	21	00	F0	06	07
4C10	36	02	23	10	FB	06	07	36
4C18	04	23	10	FB	06	0B	36	06
4C20	23	10	FB	06	0A	36	02	23
4C28	10	FB	06	05	36	04	23	10
4C30	FB	21	50	F0	06	02	3E	01
4C38	C5	06	03	77	23	10	FC	06
4C40	21	36	04	23	10	FB	06	04
4C48	77	23	10	FC	C1	10	E9	06
4C50	02	C5	06	03	77	23	10	FC
4C58	06	21	36	06	23	10	FB	06
4C60	04	77	23	10	FC	C1	10	E9
4C68	06	02	C5	06	03	77	23	10
4C70	FC	06	21	36	02	23	10	FB
4C78	06	04	77	23	10	FC	C1	10
4C80	E9	06	10	C5	06	03	77	23
4C88	10	FC	06	21	36	03	23	10
4C90	FB	06	04	77	23	10	FC	C1
4C98	10	E9	3E	21	32	98	42	00
4CA0	06	27	36	09	23	10	FB	C9
4CA8	3A	88	42	11	70	47	6F	26
4CB0	00	19	3A	89	42	E6	01	4F
4CB8	07	07	81	C6	A0	06	05	77
4CC0	23	3C	10	FB	11	23	00	19
4CC8	C6	0B	06	05	77	23	3C	10
4CD0	FB	11	D3	AB	19	36	06	23
4CD8	06	03	36	03	23	10	FB	36
4CE0	06	11	24	00	19	36	0C	23
4CE8	06	03	36	0A	23	10	FB	36

4CF0	0C	C9	3A	8D	42	6F	26	00
4CF8	29	29	29	E5	29	29	D1	19
4D00	3A	8B	42	5F	16	44	19	3A
4D08	8E	42	3C	32	8E	42	0F	0F
4D10	0F	E6	03	C6	80	77	11	28
4D18	00	19	C6	10	77	01	D8	AB
4D20	09	36	06	19	36	0E	C9	00
4D28	3A	88	42	C6	70	6F	26	47
4D30	06	05	36	20	23	10	FB	11
4D38	23	00	19	06	05	36	20	23
4D40	10	FB	C9	00	7E	23	F5	06
4D48	20	7E	E6	F1	81	12	13	23
4D50	10	F7	F1	E6	F1	81	12	C9
4D58	7E	2B	F5	06	20	7E	E6	F1
4D60	81	12	1B	2B	10	F7	F1	E6
4D68	F1	81	12	C9	3A	8F	42	3C
4D70	32	8F	42	E6	07	C0	3A	8F
4D78	42	E6	18	0F	0F	C6	04	4F
4D80	21	53	44	11	53	44	CD	44
4D88	4D	21	7B	44	11	7B	44	CD
4D90	44	4D	21	C3	44	11	C3	44
4D98	CD	58	4D	21	EB	44	11	EB
4DA0	44	CD	58	4D	21	F3	44	11
4DA8	F3	44	CD	44	4D	21	1B	45
4DB0	11	1B	45	CD	44	4D	C9	00
4DB8	2A	8A	42	ED	5B	90	42	19
4DC0	22	94	42	2A	8C	42	ED	5B
4DC8	92	42	19	22	96	42	3A	97
4DD0	42	6F	26	00	29	29	29	E5
4DD8	29	29	D1	19	3A	95	42	5F
4DE0	16	44	19	CD	EE	4D	11	28
4DE8	00	19	CD	EE	4D	C9	7E	FE
4DF0	8C	38	33	FE	8F	30	0F	2A
4DF8	90	42	7D	2F	6F	7C	2F	67
4E00	22	90	42	F1	C9	00	CA	30
4E08	4F	FE	9C	38	19	FE	B1	28
4E10	4F	FE	B2	28	55	FE	B3	28
4E18	4C	FE	B6	28	43	FE	B7	28
4E20	49	FE	B8	28	40	C9	FE	84
4E28	D8	FE	90	38	03	FE	94	D8
4E30	E6	11	B7	28	14	FE	01	20
4E38	03	2B	18	0D	FE	10	20	06
4E40	11	D8	FF	19	18	03	2B	18
4E48	F7	7E	FE	8E	28	43	23	7E
4E50	FE	8C	28	52	2B	3E	C0	77
4E58	23	77	11	27	00	19	18	2C
4E60	11	F8	FF	18	08	11	08	00
4E68	18	03	11	00	00	2A	90	42
4E70	7C	FE	01	28	05	FE	FE	28
4E78	01	19	22	90	42	2A	92	42
4E80	11	00	00	EB	ED	52	22	92
4E88	42	C3	80	4F	77	23	77	18
4E90	28	23	3E	C0	77	11	28	00

4E98	19	77	11	F8	FF	19	77	11
4EA0	28	00	19	77	18	13	3E	C0
4EA8	2B	77	11	28	00	19	77	11
4EB0	BF	FF	19	77	11	28	00	19
4EB8	77	11	00	AC	19	7E	0E	0A
4EC0	FE	02	28	08	0E	19	FE	06
4EC8	28	02	0E	32	06	00	2A	9A
4ED0	42	09	22	9A	42	E5	CD	0C
4ED8	4F	E1	ED	5B	9C	42	DF	38
4EE0	03	22	9C	42	2A	9C	42	11
4EE8	23	44	CD	0F	4F	3E	08	D3
4EF0	F8	3E	10	D3	F9	3E	0D	D3
4EF8	F8	3E	09	D3	F9	C3	20	50
4F00	3E	30	3C	09	38	FC	3D	ED
4F08	42	12	13	C9	11	07	44	01
4F10	F0	D8	CD	00	4F	01	18	FC
4F18	CD	00	4F	01	9C	FF	CD	00
4F20	4F	01	F6	FF	CD	00	4F	01
4F28	FF	FF	CD	00	4F	C9	00	00
4F30	26	47	3A	8B	42	C6	98	6F
4F38	36	9C	23	36	9D	23	36	9E
4F40	11	FE	AB	19	36	0E	23	36
4F48	0E	23	36	06	3E	09	D3	F8
4F50	3E	0F	D3	F9	06	10	3E	03
4F58	D3	F8	78	D3	F9	C5	78	E6
4F60	07	3C	47	CD	60	00	C1	10
4F68	ED	3E	09	D3	F8	AF	D3	F9
4F70	01	00	00	CD	60	00	C3	34
4F78	50	00	00	00	00	00	00	00
4F80	3E	0A	D3	F8	3E	10	D3	F9
4F88	3E	08	D3	F8	3E	00	D3	F9
4F90	3E	0D	D3	F8	3E	09	D3	F9
4F98	3E	05	D3	F8	3E	0F	D3	F9
4FA0	CD	F0	14	3A	AA	40	E6	03
4FA8	3D	3D	5F	16	00	FE	80	38
4FB0	02	16	FF	2A	90	42	19	22
4FB8	90	42	F1	C9	CD	00	4C	21
4FC0	03	00	22	88	42	21	00	06
4FC8	22	8A	42	21	00	08	22	8C
4FD0	42	21	20	00	22	90	42	21
4FD8	00	00	22	92	42	C9	00	00
4FE0	21	50	F0	06	02	C5	23	23
4FE8	23	06	21	36	04	23	10	FB
4FF0	23	23	23	23	C1	10	EE	06
4FF8	02	C5	06	21	23	23	23	36
5000	06	23	10	FB	23	23	23	23
5008	C1	10	EE	06	02	C5	06	21
5010	23	23	23	36	02	23	10	FB
5018	23	23	23	23	C1	10	EE	C9
5020	2A	81	56	7C	B7	28	04	2B
5028	22	81	56	F1	3A	98	42	3D
5030	32	98	42	C9	3A	16	48	3D
5038	32	16	48	FE	30	28	05	CD
5040	BC	4F	F1	C9	F1	F1	C3	00
5048	57	00	00	00	00	00	00	00
5050	00	00	00	00	00	00	00	00
5058	00	00	00	00	00	00	00	00
5060	00	00	00	00	00	00	00	00
5068	00	00	00	00	00	00	00	00
5070	00	00	00	00	00	00	00	00
5078	00	00	00	00	00	00	00	00

5080	00	00	00	00	00	00	00	00
5088	00	00	00	00	00	00	00	00
5090	00	00	00	00	00	00	00	00
5098	00	00	00	00	00	00	00	00
50A0	00	00	00	00	00	00	00	00
50A8	00	00	00	00	00	00	00	00
50B0	00	00	00	00	00	00	00	00
50B8	00	00	00	00	00	00	00	00
50C0	00	00	00	00	00	00	00	00
50C8	00	00	00	00	00	00	00	00
50D0	00	00	00	00	00	00	00	00
50D8	00	00	00	00	00	00	00	00
50E0	00	00	00	00	00	00	00	00
50E8	00	00	00	00	00	00	00	00
50F0	00	00	00	00	00	00	00	00
50F8	00	00	00	00	00	00	00	00
5100	8C	8D	8E	20	8F	8F	8F	20
5108	20	20	8F	8F	20	20	8F	20
5110	20	20	20	8F	20	20	20	20
5118	20	8F	8F	20	20	8F	20	20
5120	8F	20	20	20	8C	8D	8E	20
5128	8C	8D	8E	20	8F	20	20	8F
5130	20	8F	20	20	8F	20	8F	20
5138	20	20	20	8F	20	20	20	20
5140	8F	20	20	8F	20	8F	8F	20
5148	8F	20	20	20	8C	8D	8E	20
5150	8C	8D	8E	20	8F	8F	8F	20
5158	20	8F	8F	8F	8F	20	8F	20
5160	20	20	20	8F	20	20	20	20
5168	8F	20	20	8F	20	8F	20	8F
5170	8F	20	20	20	8C	8D	8E	20
5178	8C	8D	8E	20	8F	20	20	8F
5180	20	8F	20	20	8F	20	8F	20
5188	20	20	20	8F	20	20	20	20
5190	8F	20	20	8F	20	8F	20	20
5198	8F	20	20	20	8C	8D	8E	20
51A0	8C	8D	8E	20	8F	8F	8F	20
51A8	20	8F	20	20	8F	20	8F	8F
51B0	8F	8F	20	8F	8F	8F	8F	20
51B8	20	8F	8F	20	20	8F	20	20
51C0	8F	20	20	20	8C	8D	8E	20
51C8	8C	8D	8E	20	20	20	20	20
51D0	20	20	20	20	20	20	20	20
51D8	20	20	20	20	20	20	20	20
51E0	20	20	20	20	20	20	20	20
51E8	20	20	20	20	8C	8D	8E	20
51F0	20	20	20	20	20	20	20	20
51F8	20	20	20	20	20	20	20	20
5200	3C	7A	73	7E	3C	19	3D	7F
5208	3C	7E	7E	7E	BD	99	BD	FF
5210	3C	0E	0E	7E	BC	98	BC	FE
5218	3C	7E	7E	7E	3C	18	3C	7E
5220	03	0F	1F	31	27	27	27	11
5228	C0	F0	F8	8C	3C	9C	CC	18
5230	03	0F	1F	21	33	33	33	13
5238	C0	F0	F8	8C	3C	3C	3C	88
5240	03	0F	1F	3F	3F	3F	3F	1F
5248	C0	F0	F8	84	CC	CC	CC	C8
5250	03	0F	1F	30	27	31	3C	11
5258	C0	F0	F8	FC	FC	FC	FC	F8
5260	07	07	03	03	03	03	03	03

5268	FF	FF	00	00	00	00	00	00
5270	E0	E0	C0	C0	C0	C0	C0	C0
5278	E7	E7	00	7E	7E	00	E7	E7
5280	FE	B0	8C	3E	76	67	60	E0
5288	FE	00	3C	7E	66	E7	00	00
5290	7F	0D	31	7D	6E	E6	06	07
5298	FF	BD	81	BD	7E	66	66	E7
52A0	0F	03	06	0F	00	00	00	00
52A8	F0	C0	00	00	00	00	00	00
52B0	0F	03	01	03	00	00	00	00
52B8	F0	C0	80	C0	00	00	00	00
52C0	0F	03	00	00	00	00	00	00
52C8	F0	C0	60	F0	00	00	00	00
52D0	0F	03	01	03	00	00	00	00
52D8	F0	C0	80	C0	00	00	00	00
52E0	00	00	00	00	81	83	FF	FF
52E8	00	00	00	00	7C	7F	03	7F
52F0	00	00	00	08	3C	76	F6	BC
52F8	00	00	00	00	00	00	00	00
5300	00	3C	7B	7E	3C	18	3C	76
5308	00	00	00	00	00	00	00	00
5310	00	00	00	00	00	00	00	00
5318	00	00	00	00	00	00	00	00
5320	00	3C	DE	7E	3C	18	3C	6E
5328	04	78	EC	FC	78	30	78	DC
5330	00	00	00	00	00	00	00	00
5338	00	00	00	00	00	00	00	00
5340	00	00	00	00	00	00	00	00
5348	20	1E	37	3F	1E	0C	1E	3B
5350	00	00	00	00	00	00	00	00
5358	00	00	00	00	00	00	00	00
5360	00	00	00	00	00	00	00	00
5368	00	00	00	00	00	00	00	00
5370	00	00	00	00	00	00	00	00
5378	00	00	00	00	00	00	00	00
5380	7B	7D	7C	3E	36	36	36	3F
5388	F0	0F	00	00	00	00	00	00
5390	00	00	FF	00	00	00	00	00
5398	0F	F0	00	00	00	00	00	00
53A0	DE	BE	3E	7C	6C	6C	6C	FC
53A8	EF	F6	78	30	59	6F	66	70
53B0	FF	00	00	00	00	00	00	00
53B8	FF	00	00	00	00	00	00	00
53C0	FF	00	00	00	00	00	00	00
53C8	F7	6F	1E	0C	9A	F6	66	0E
53D0	00	00	00	00	00	00	00	00
53D8	00	00	00	00	00	00	00	00
53E0	00	00	00	00	00	00	00	00
53E8	00	00	00	00	00	00	00	00
53F0	00	00	00	00	00	00	00	00
53F8	00	00	00	00	00	00	00	00
5400	00	00	00	00	00	00	00	00
5408	00	00	00	00	00	00	00	00
5410	00	00	00	00	00	00	00	00
5418	00	00	00	00	00	00	00	00
5420	00	00	00	00	00	00	00	00
5428	00	00	00	00	00	00	00	00
5430	00	00	00	00	00	00	00	00
5438	00	00	00	00	00	00	00	00
5440	00	00	00	00	00	00	00	00
5448	00	00	00	00	00	00	00	00

5450	00	00	00	00	00	00	00	00
5458	00	00	00	00	00	00	00	00
5460	00	00	00	00	00	00	00	00
5468	00	00	00	00	00	00	00	00
5470	00	00	00	00	00	00	00	00
5478	00	00	00	00	00	00	00	00
5480	00	00	00	00	00	00	00	00
5488	00	00	00	00	00	00	00	00
5490	00	00	00	00	00	00	00	00
5498	00	00	00	00	00	00	00	00
54A0	00	00	00	00	00	00	00	00
54A8	00	00	00	00	00	00	00	00
54B0	00	00	00	00	00	00	00	00
54B8	00	00	00	00	00	00	00	00
54C0	00	00	00	00	00	00	00	00
54C8	00	00	00	00	00	00	00	00
54D0	00	00	00	00	00	00	00	00
54D8	00	00	00	00	00	00	00	00
54E0	00	00	00	00	00	00	00	00
54E8	00	00	00	00	00	00	00	00
54F0	00	00	00	00	00	00	00	00
54F8	00	00	00	00	00	00	00	00
5500	00	00	00	00	00	00	00	00
5508	00	00	00	00	00	00	00	00
5510	00	00	00	00	00	00	00	00
5518	00	00	00	00	00	00	00	00
5520	00	00	00	00	00	00	00	00
5528	00	00	00	00	00	00	00	00
5530	00	00	00	00	00	00	00	00
5538	00	00	00	00	00	00	00	00
5540	00	00	00	00	00	00	00	00
5548	00	00	00	00	00	00	00	00
5550	00	00	00	00	00	00	00	00
5558	00	00	00	00	00	00	00	00
5560	00	00	00	00	00	00	00	00
5568	00	00	00	00	00	00	00	00
5570	00	00	00	00	00	00	00	00
5578	00	00	00	00	00	00	00	00
5580	00	00	00	00	00	00	00	00
5588	00	00	00	00	00	00	00	00
5590	00	00	00	00	00	00	00	00
5598	00	00	00	00	00	00	00	00
55A0	00	00	00	00	00	00	00	00
55A8	00	00	00	00	00	00	00	00
55B0	00	00	00	00	00	00	00	00
55B8	00	00	00	00	00	00	00	00
55C0	00	00	00	00	00	00	00	00
55C8	00	00	00	00	00	00	00	00
55D0	00	00	00	00	00	00	00	00
55D8	00	00	00	00	00	00	00	00
55E0	00	00	00	00	00	00	00	00
55E8	00	00	00	00	00	00	00	00
55F0	00	00	00	00	00	00	00	00
55F8	00	00	00	00	00	00	00	00
5600	CD	28	4D	ED	4B	88	42	3A
5608	40	F8	CB	6F	28	01	0D	CB
5610	77	28	01	0C	79	FE	03	30
5618	01	3C	FE	20	38	01	3D	4F
5620	3A	88	42	B9	28	01	04	ED
5628	43	88	42	CD	A8	4C	3A	8D
5630	42	6F	26	00	29	29	29	E5
5638	29	29	D1	19	3A	8B	42	5F

5640	16	44	19	36	C0	11	28	00
5648	19	36	C0	CD	6C	4D	CD	B8
5650	4D	2A	90	42	ED	5B	8A	42
5658	19	22	8A	42	2A	92	42	ED
5660	5B	8C	42	19	22	8C	42	2A
5668	92	42	7C	FE	01	28	01	23
5670	22	92	42	CD	E0	4F	CD	F2
5678	4C	00	00	00	00	00	00	00
5680	01	00	06	CD	60	00	3A	98
5688	42	B7	20	03	CD	BC	4F	3A
5690	40	F8	FE	06	C2	00	56	C3
5698	00	57	00	00	00	00	00	00
56A0	00	00	00	00	00	00	00	00
56A8	00	00	00	00	00	00	00	00
56B0	00	00	00	00	00	00	00	00
56B8	00	00	00	00	00	00	00	00
56C0	00	00	00	00	00	00	00	00
56C8	00	00	00	00	00	00	00	00
56D0	00	00	00	00	00	00	00	00
56D8	00	00	00	00	00	00	00	00
56E0	00	00	00	00	00	00	00	00
56E8	00	00	00	00	00	00	00	00
56F0	00	00	00	00	00	00	00	00
56F8	00	00	00	00	00	00	00	00
5700	CD	C9	01	21	03	00	22	88
5708	42	21	00	06	22	8A	42	21
5710	00	08	22	8C	42	21	30	00
5718	22	90	42	21	00	00	22	92
5720	42	21	00	00	22	9A	42	00
5728	00	00	3E	07	D3	F8	3E	F1
5730	D3	F9	3E	08	D3	F8	AF	D3
5738	F9	3E	09	D3	F8	AF	D3	F9
5740	3E	0A	D3	F8	AF	D3	F9	21
5748	00	06	22	81	56	21	00	52
5750	11	00	F4	01	00	04	ED	B0
5758	3E	00	CD	B0	3F	3E	36	32
5760	16	48	CD	00	4C	21	00	51
5768	11	80	46	01	F0	00	ED	B0
5770	2A	98	42	CD	0C	4F	2A	9A
5778	42	11	23	44	CD	0F	4F	CD
5780	49	00	CD	00	4C	C3	00	56
5788	00	00	00	00	00	00	00	00
5790	21	00	00	22	98	42	C3	00
5798	57	00	00	00	00	00	00	00
57A0	00	00	00	00	00	00	00	00
57A8	00	00	00	00	00	00	00	00
57B0	00	00	00	00	00	00	00	00
57B8	00	00	00	00	00	00	00	00
57C0	00	00	00	00	00	00	00	00
57C8	00	00	00	00	00	00	00	00
57D0	00	00	00	00	00	00	00	00
57D8	00	00	00	00	00	00	00	00
57E0	00	00	00	00	00	00	00	00
57E8	00	00	00	00	00	00	00	00
57F0	00	00	00	00	00	00	00	00
57F8	00	00	00	00	00	00	00	00

Der Zeichensatz des Programms BALLON

CHR$(128)	CHR$(129)	CHR$(130)	CHR$(131)	CHR$(132)	CHR$(133)
CHR$(134)	CHR$(135)	CHR$(136)	CHR$(137)	CHR$(138)	CHR$(139)
CHR$(140)	CHR$(141)	CHR$(142)	CHR$(143)	CHR$(144)	CHR$(145)
CHR$(146)	CHR$(147)	CHR$(148)	CHR$(149)	CHR$(150)	CHR$(151)
CHR$(152)	CHR$(153)	CHR$(154)	CHR$(155)	CHR$(156)	CHR$(157)
CHR$(158)	CHR$(159)	CHR$(160)	CHR$(161)	CHR$(162)	CHR$(163)

CHR$(164)	CHR$(165)	CHR$(166)	CHR$(167)	CHR$(168)	CHR$(169)
CHR$(170)	CHR$(171)	CHR$(172)	CHR$(173)	CHR$(174)	CHR$(175)
CHR$(176)	CHR$(177)	CHR$(178)	CHR$(179)	CHR$(180)	CHR$(181)
CHR$(182)	CHR$(183)	CHR$(184)	CHR$(185)	CHR$(186)	CHR$(187)
CHR$(188)	CHR$(189)	CHR$(190)	CHR$(191)	CHR$(192)	CHR$(193)

Ein leistungsfähiger Maschinensprache-Monitor

Der Monitor, den Sie auf den folgenden Seiten finden, bietet nicht nur viele leistungsstarke Befehle, er hat noch einen anderen großen Vorteil: Er ist so optimiert programmiert, daß er nur 4K Byte lang ist.
Damit ist er schnell geladen und nimmt nur wenig Speicherplatz in Anspruch.
Ferner ist der Monitor voll relozierbar.
Wenn Sie die Moglichkeit haben, 2532-Eproms zu programmieren, konnen Sie diesen Monitor sogar im EPROM laufen lassen !!
Dies ist besonders interessant fur Diskettenbenutzer, die ja in der DOS-Cartridge einen Epromsteckplatz (E000H-EFFFH) frei haben.

Im folgenden finden Sie zwei Hex-Listings: Das erste ist die Diskettenversion des Monitors (A800H-B7FFH), das zweite die Kassettenversion (7000H-7FFFH).
Die Startadressen sind A800H bzw. 7000H.

Die Befehle des Monitors:

Jeder Befehl des Monitors hat folgendes Format:
Xaaaa< bbbb< cccc>> RETURN, wobei X einer der im folgenden beschriebenen Befehlskurzel und aaaa bis cccc Hexadezimalzahlen sind.

Aaaaa Ascii-Edit
Die Pfeiltasten bewegen den Cursor,
<BREAK> kehrt zur Kommandoeingabe zuruck,
<CLEAR> schaltet um zum Hex-Edit.

Da Drive-Select (nur Diskversion!!)
Das Laufwerk a (0-3) wird fur den nachfolgenden Zugriff selektiert.

E Exit-Basic

Faaaa bb Find Bytes
Sucht ab Adresse aaaa nach maximal 4 Bytes bb.
Werden keine Bytes angegeben, so wird mit den zuletzt angegebenen Bytes ab der neuen Adresse weitergesucht, wird auch die Adresse weggelassen, so wird mit den alten Bytes ab der alten Adresse+1 weitergesucht Wird die Bytefolge im Speicher gefunden, so werden Hex-Dump und Disassembler auf die gefundene Adresse gesetzt.

Gaaaa Go Address
Das Programm, das bei Adresse aaaa beginnt, wird als Unterprogramm aufgerufen, sodaß mit einem RET-Befehl in den Monitor zuruckgekehrt werden kann. Die Register werden so ubernommen, wie sie in der Anzeige stehen, hinterher werden die neuen Inhalte angezeigt.

Haaaa Hex-Edit
Pfeiltasten bewegen den Cursor,
<BREAK> kehrt zur Befehlseingabe zuruck,
<CLEAR> schaltet um auf Ascii-Edit,
<;>, <-> schalten eine Seite (64 Bytes) vor/zuruck
<F1> markiert Cursorstelle
<F2> berechnet relatives Offset von der Cursorposition zur letzten mit <F1> abgespeicherten Adresse, setzt dieses in die Adresse <F1> ein.

Ia Inhaltsverzeichnis (nur Diskversion!)
Das Inhaltsverzeichnis der Diskette in Laufwerk a wird angezeigt.

L Load
Ein Programm von Kassette/Diskette laden.
Startadresse, Endadresse und Einsprungadresse werden angezeigt.

Maaaa bbbb cccc
Move memory
Verschiebt den Speicherbereich von Adresse aaaa bis Adresse bbbb auf Adresse cccc. Die Bereiche durfen sich auch uberlappen.

Oaaaa bbbb cccc dddd eeee
Offset berechnen
Im Speicherbereich von aaaa bis bbbb werden alle Adressen, die zwischen cccc und dddd liegen, in Adressen ab eeee umgeandert (Relozierfunktion).

Paaaa bbbb
Printerlisting
Der Speicherbereich von Adresse aaaa bis bbbb wird disassembliert auf den Drucker ausgegeben.

Qaaaa Disassembler
Ab Adresse aaaa werden 11 Zeilen disassembliert ausgegeben. Mit dem Aufwarts-/Abwartspfeil konnen Sie einen Befehl ruckwarts/vorwarts gehen, mit <->,<;> 11 Befehle (1 Seite).

Rsss aaaa cc
Read sectors (nur Diskettenversion!)
Von der mit D selektierten Diskette werden cc Sektoren ab Sektor sss in den Speicher ab Adresse aaaa geladen.

Saaaa bbbb cccc
Save memory
Schreibt den Speicherbereich von aaaa bis bbbb mit der Einsprungadresse cccc auf Kassette/Diskette.

V View file

Ein Programm auf Kassette/Diskette wird auf seine Ladeadressen hin überprüft. Auf diese Art und Weise können Sie feststellen, wohin ein Programm geladen würde, wenn es geladen würde.

Wsss aaaa cc

Write sectors (nur Diskettenversion!)

Schreibt den Speicherbereich ab Adresse aaaa auf die momentan selektierte Diskette ab Sektor sss. cc Sektoren werden beschrieben.

Xaaaa bbbb

Xecute calculation

Berechnet Summe und Differenz von aaaa und bbbb.

Zaaaa bbbb cc

Zero memory

Füllt den Speicherbereich zwischen aaaa und bbbb mit dem Byte cc.

Die Hexlistings der beiden Monitorversionen (Disk- und Tapeversion) beginnen auf der nächsten Seite:

korrigiert 140784

```
A800  C3 89 B3 F5 F5 0F 0F 0F
A808  0F CD 26 A8 F1 CD 26 A8
A810  F1 C9 FE A0 38 06 F5 3E
A818  30 12 13 F1 F5 CD 03 A8
A820  3E 48 12 13 F1 C9 E6 0F
A828  C6 90 27 CE 40 27 12 13
A830  C9 F5 7C FE A0 38 05 3E
A838  30 12 13 7C CD 03 A8 7D
A840  18 DB 7C CD 03 A8 7D C3
A848  03 A8 C5 F5 2A 00 42 7E
A850  FE ED 20 01 23 23 4E 23
A858  46 C5 E1 CD 31 A8 CD EE
A860  B6 F1 C1 C9 F5 3E 28 12
A868  13 F1 C9 F5 3E 29 18 F7
A870  F5 3E 2C 18 F2 F5 3E 41
A878  18 ED CD 75 A8 18 F1 CD
A880  70 A8 18 F1 CD 87 A8 F5
A888  3E 20 18 DB F5 3E 2B 18
A890  D6 3E 02 18 02 3E 03 C3
A898  F4 A8 E6 38 0F 4F 06 00
A8A0  C9 C5 E5 F5 E6 07 4F 06
A8A8  00 21 68 AE 09 7E FE 4D
A8B0  28 06 12 13 F1 E1 C1 C9
A8B8  3A 05 42 B7 20 0D CD 64
A8C0  A8 CD 91 A8 CD 6B A8 F1
A8C8  E1 C1 C9 C6 03 CD 64 A8
A8D0  CD F4 A8 CD 8C A8 2A 00
A8D8  42 23 7E CD 12 A8 CD 6B
A8E0  A8 F1 E1 C1 F5 3A 06 42
A8E8  B7 28 07 AF 32 06 42 04
A8F0  04 23 F1 C9 C5 E5 F5 E6
A8F8  07 07 4F 06 00 21 70 AE
A900  09 7E 12 13 23 7E 12 13
A908  FE 4C 28 04 F1 E1 C1 C9
A910  3A 05 42 B7 28 F6 1B 1B
A918  3E 49 12 13 3A 05 42 C6
A920  57 12 13 F1 E1 C1 F5 3A
A928  06 42 B7 28 05 AF 32 06
A930  42 04 F1 C9 C5 01 04 00
A938  ED B0 CD 87 A8 C1 C9 F5
A940  E6 38 0F 0F 0F CD A1 A8
A948  F1 C9 C5 F5 2A 00 42 23
A950  7E 4F 06 00 FE 80 38 02
A958  06 FF 23 09 CD 31 A8 F1
A960  C1 C9 F5 E6 30 0F 0F 0F
A968  0F CD F4 A8 F1 C9 CD 62
A970  A9 F5 1B 1A FE 50 20 07
A978  1B CD 75 A8 3E 46 12 13
A980  F1 C9 C5 CD 9A A8 CB 09
A988  21 7C AE 09 7E 12 23 13
A990  7E FE 20 28 02 12 13 C1
```

Korrigiert 14.07.84

```
A998  C9 D6 20 CD 82 A9 CD 70
A9A0  A8 C3 4A A9 F5 E6 07 4F
A9A8  0C F1 0D CA 7E AA 0D CA
A9B0  56 AA 0D 28 51 0D 28 3A
A9B8  0D 28 32 0D 28 24 0D 28
A9C0  0C 21 D8 AE CD 9A A8 09
A9C8  06 01 C3 34 A9 E5 21 38
A9D0  AF 06 02 CD 34 A9 E1 CD
A9D8  3F A9 CD 70 A8 23 7E C3
A9E0  12 A8 21 80 AF CD 34 A9
A9E8  06 01 C3 3F A9 21 7C AF
A9F0  18 F3 CB 5F 20 05 21 7C
A9F8  AF 18 03 21 80 AF CD 34
AA00  A9 06 01 C3 62 A9 21 38
AA08  AF CD 34 A9 FE 20 30 18
AA10  CB 5F 28 0E CD 7A A8 CD
AA18  64 A8 06 01 CD 62 A9 C3
AA20  6B A8 CD 17 AA C3 7F A8
AA28  06 03 CB 5F 28 18 FE 30
AA30  38 05 CD 7A A8 18 06 CD
AA38  91 A8 CD 70 A8 CD 64 A8
AA40  CD 4A A8 C3 6B A8 06 03
AA48  CD 3D AA CD 70 A8 FE 30
AA50  D2 75 A8 C3 91 A8 CB 5F
AA58  28 13 21 8C AE CD 34 A9
AA60  06 01 F5 CD 91 A8 F1 CD
AA68  70 A8 C3 62 A9 21 38 AF
AA70  CD 34 A9 06 03 CD 62 A9
AA78  CD 70 A8 C3 4A A8 B7 20
AA80  06 21 94 AF C3 E1 AB FE
AA88  08 20 18 21 88 AF CD 34
AA90  A9 3E 30 CD 6E A9 CD 70
AA98  A8 CD 6E A9 3E 27 12 13
AAA0  06 01 C9 FE 10 20 0B 21
AAA8  58 AF CD 34 A9 06 02 C3
AAB0  4A A9 21 5C AF CD 34 A9
AAB8  FE 18 28 F1 06 02 C3 99
AAC0  A9 FE 76 20 08 21 6C AF
AAC8  06 01 C3 34 A9 21 38 AF
AAD0  CD 34 A9 06 01 CD 3F A9
AAD8  CD 70 A8 C3 A1 A8 F5 CD
AAE0  9A A8 21 8C AE 09 CD 34
AAE8  A9 FE 08 28 07 FE 10 30
AAF0  03 CD 7A A8 06 01 F1 C3
AAF8  A1 A8 F5 E6 07 4F 0C F1
AB00  0D CA 3B AC 0D CA F4 AB
AB08  0D CA EB AB 0D 28 5C 0D
AB10  28 48 0D 28 2E 0D 28 0D
AB18  21 60 AF CD 34 A9 06 01
AB20  E6 38 C3 12 A8 CD 9A A8
AB28  E5 21 8C AE 09 CD 34 A9
```

15.03.84

```
AB30  FE 08 28 07 FE 10 30 03
AB38  CD 7A A8 06 02 E1 23 7E
AB40  C3 12 A8 CB 5F 28 08 21
AB48  50 AF CD 34 A9 18 17 21
AB50  54 AF CD 34 A9 06 01 C3
AB58  6E A9 21 50 AF CD 34 A9
AB60  CD 82 A9 CD 70 A8 06 03
AB68  C3 4A A8 FE C3 20 08 21
AB70  4C AF CD 34 A9 18 EF FE
AB78  DB 20 19 21 3C AF CD 34
AB80  A9 CD 7A A8 2A 00 42 23
AB88  CD 64 A8 7E CD 12 A8 06
AB90  02 C3 6B A8 FE D3 20 0C
AB98  21 40 AF CD 34 A9 CD 84
ABA0  AB C3 7F A8 FE E3 20 17
ABA8  21 88 AF CD 34 A9 21 90
ABB0  AF CD 34 A9 1B CD 70 A8
ABB8  06 01 3E 20 C3 62 A9 FE
ABC0  EB 20 17 21 88 AF CD 34
ABC8  A9 3E 01 CD F4 A8 CD 70
ABD0  A8 AF 32 05 42 06 01 C3
ABD8  91 A8 FE F3 20 08 21 64
ABE0  AF 06 01 C3 34 A9 21 68
ABE8  AF 18 F6 21 4C AF CD 34
ABF0  A9 C3 60 AB CB 5F 28 38
ABF8  FE D9 20 05 21 8C AF 18
AC00  E0 FE C9 20 05 21 44 AF
AC08  18 D7 FE E9 20 11 21 4C
AC10  AF CD 34 A9 CD 64 A8 06
AC18  01 CD 91 A8 C3 6B A8 21
AC20  38 AF CD 34 A9 06 01 CD
AC28  95 A8 CD 70 A8 C3 91 A8
AC30  21 48 AF CD 34 A9 06 01
AC38  C3 6E A9 21 44 AF CD 34
AC40  A9 06 01 C3 82 A9 3A 05
AC48  42 B7 28 01 23 23 7E FE
AC50  40 38 25 F5 E6 C0 07 07
AC58  07 07 4F 06 00 21 C8 AE
AC60  09 CD 34 A9 F1 F5 E6 38
AC68  0F 0F 0F C6 30 12 13 F1
AC70  CD 70 A8 06 02 C3 A1 A8
AC78  F5 CD 9A A8 F1 21 AC AE
AC80  09 CD 34 A9 06 02 C3 A1
AC88  A8 23 7E FE C0 30 7A FE
AC90  A0 D2 CA AD FE 80 30 71
AC98  FE 40 38 6D F5 E6 07 4F
ACA0  F1 0C 0D CA 98 AD 0D CA
ACA8  81 AD 0D CA B0 AD 0D CA
ACB0  63 AD 0D CA 57 AD 0D CA
ACB8  3D AD 0D CA 1B AD FE 70
ACC0  30 47 FE 60 30 2B 21 38
```

ACC8	AF	CD	34	A9	CB	5F	28	11
ACD0	CD	7A	A8	06	49	FE	5F	20
ACD8	02	06	52	78	12	13	06	02
ACE0	C9	06	49	FE	4F	20	02	06
ACE8	52	78	12	13	06	02	C3	7F
ACF0	A8	21	74	AF	CB	5F	28	03
ACF8	21	78	AF	CD	34	A9	06	02
AD00	CD	64	A8	CD	91	A8	C3	6B
AD08	A8	21	A0	AF	11	E8	41	CD
AD10	34	A9	2A	00	42	7E	06	01
AD18	C3	12	A8	FE	60	30	EA	FE
AD20	4E	28	E6	21	84	AF	CD	34
AD28	A9	06	30	FE	46	28	08	06
AD30	31	FE	56	28	02	06	32	78
AD38	12	13	06	02	C9	FE	50	30
AD40	C8	21	44	AF	CD	34	A9	1B
AD48	1B	06	49	FE	4D	28	02	06
AD50	4E	78	12	13	06	02	C9	FE
AD58	44	20	AE	21	70	AF	06	02
AD60	C3	34	A9	21	38	AF	CD	34
AD68	A9	06	04	CB	5F	20	09	CD
AD70	3D	AA	CD	70	A8	C3	62	A9
AD78	CD	62	A9	CD	70	A8	C3	3D
AD80	AA	FE	71	CA	09	AD	21	40
AD88	AF	CD	34	A9	21	98	AF	CD
AD90	34	A9	1B	06	02	C3	3F	A9
AD98	FE	70	CA	09	AD	21	3C	AF
ADA0	CD	34	A9	06	02	CD	3F	A9
ADA8	21	9C	AF	CD	34	A9	1B	C9
ADB0	21	90	AE	CB	5F	20	03	21
ADB8	98	AE	CD	34	A9	F5	CD	91
ADC0	A8	F1	CD	70	A8	06	02	C3
ADC8	62	A9	E6	07	FE	04	D2	09
ADD0	AD	E6	03	07	07	07	07	4F
ADD8	7E	E6	18	0F	81	4F	06	00
ADE0	21	F8	AE	09	06	02	C3	34
ADE8	A9	22	00	42	C5	D5	11	E8
ADF0	41	01	55	AE	C5	AF	32	05
ADF8	42	32	06	42	7E	FE	40	DA
AE00	A4	A9	FE	80	DA	C1	AA	FE
AE08	C0	DA	DE	AA	FE	CB	CA	46
AE10	AC	FE	ED	CA	89	AC	FE	DD
AE18	20	22	3E	01	32	05	42	32
AE20	06	42	23	22	00	42	01	46
AE28	AE	C5	7E	FE	DD	CA	45	AE
AE30	FE	FD	CA	45	AE	FE	ED	CA
AE38	45	AE	18	C0	FE	FD	C2	FA
AE40	AA	3E	02	18	D7	C1	2A	00
AE48	42	2B	22	00	42	3A	06	42
AE50	B7	C2	09	AD	C9	AF	12	78
AE58	32	04	42	48	06	00	2A	00

16.03.87

AE60	42	09	22	02	42	D1	C1	C9
AE68	42	43	44	45	48	4C	4D	41
AE70	42	43	44	45	48	4C	53	50
AE78	49	58	49	59	4E	5A	5A	20
AE80	4E	43	43	20	50	4F	50	45
AE88	50	20	4D	20	41	44	44	20
AE90	41	44	43	20	53	55	42	20
AE98	53	42	43	20	41	4E	44	20
AEA0	58	4F	52	20	4F	52	20	20
AEA8	43	50	20	20	52	4C	43	20
AEB0	52	52	43	20	52	4C	20	20
AEB8	52	52	20	20	53	4C	41	20
AEC0	53	52	41	20	53	4C	49	20
AEC8	53	52	4C	20	42	49	54	20
AED0	52	45	53	20	53	45	54	20
AED8	52	4C	43	41	52	52	43	41
AEE0	52	4C	41	20	52	52	41	20
AEE8	44	41	41	20	43	50	4C	20
AEF0	53	43	46	20	43	43	46	20
AEF8	4C	44	49	20	4C	44	44	20
AF00	4C	44	49	52	4C	44	44	52
AF08	43	50	49	20	43	50	44	20
AF10	43	50	49	52	43	50	44	52
AF18	49	4E	49	20	49	4E	44	20
AF20	49	4E	49	52	49	4E	44	52
AF28	4F	55	54	49	4F	55	54	44
AF30	4F	54	49	52	4F	54	44	52
AF38	4C	44	20	20	49	4E	20	20
AF40	4F	55	54	20	52	45	54	20
AF48	50	4F	50	20	4A	50	20	20
AF50	43	41	4C	4C	50	55	53	48
AF58	44	4A	4E	5A	4A	52	20	20
AF60	52	53	54	20	44	49	20	20
AF68	45	49	20	20	48	41	4C	54
AF70	4E	45	47	20	52	52	44	20
AF78	52	4C	44	20	49	4E	43	20
AF80	44	45	43	20	49	4D	20	20
AF88	45	58	20	20	45	58	58	20
AF90	28	53	50	29	4E	4F	50	20
AF98	28	43	29	2C	2C	28	43	29
AFA0	44	45	46	42	11	C8	44	CD
AFA8	E9	AD	D5	C5	D5	E1	13	01
AFB0	27	00	36	20	ED	B0	C1	D1
AFB8	2A	00	42	C5	D5	CD	42	A8
AFC0	CD	87	A8	CD	87	A8	3A	04
AFC8	42	47	7E	23	CD	03	A8	10
AFD0	F9	D1	EB	01	10	00	09	EB
AFD8	D5	21	E8	41	7E	B7	28	05
AFE0	12	13	23	18	F7	E1	01	18
AFE8	00	09	C1	EB	2A	02	42	10
AFF0	B6	C9	42	43	3A	20	44	45

11.07.94

```
AFF8   3A 20 48 4C 3A 20 41 46
B000   3A 20 42 43 60 20 44 45
B008   60 20 48 4C 60 20 41 46
B010   60 20 49 58 3A 20 49 59
B018   3A 20 53 50 3A 20 50 43
B020   3A 20 20 28 48 4C 29 3A
B028   20 20 20 28 53 50 29 3A
B030   20 53 5A 58 48 58 50 4E
B038   43 46 49 4C 45 4E 41 4D
B040   45 20 53 54 41 52 54 20
B048   45 4E 44 20 20 20 45 4E
B050   54 52 59 58 20 20 20 20
B058   59 20 20 20 20 58 2B 59
B060   20 20 59 2D 58 20 20 58
B068   2D 59 21 00 F0 01 28 00
B070   11 01 F0 36 03 ED B0 06
B078   0C C5 06 04 36 02 23 10
B080   FB 06 06 36 04 23 10 FB
B088   C1 10 EE 06 02 36 02 23
B090   10 FB 06 0C 36 09 23 10
B098   FB 06 02 C5 06 06 36 02
B0A0   23 10 FB 06 04 36 04 23
B0A8   10 FB C1 10 EE E5 D1 13
B0B0   36 05 01 BE 01 ED B0 06
B0B8   08 C5 06 06 36 02 23 10
B0C0   FB 06 04 36 09 23 36 09
B0C8   23 23 36 01 23 36 01 23
B0D0   23 10 F0 06 0A 36 04 23
B0D8   10 FB C1 10 DC C9 C5 D5
B0E0   E5 11 80 46 06 08 C5 CD
B0E8   42 A8 CD 84 A8 E5 06 08
B0F0   7E CD 03 A8 CD 87 A8 23
B0F8   10 F6 E1 CD 84 A8 01 08
B100   00 ED B0 C1 10 E0 E1 D1
B108   C1 C9 CD 2B 00 B7 28 FA
B110   FE 01 CA A3 B3 C9 C5 D5
B118   E5 21 C8 44 11 C9 44 01
B120   B7 01 36 20 ED B0 E1 D1
B128   C1 C9 E5 D1 01 F8 FF 09
B130   CD E9 AD DF 38 FA 2A 00
B138   42 C9 CD 2A B1 06 0B C3
B140   A4 AF 2A 46 42 11 F8 FF
B148   18 06 2A 46 42 11 08 00
B150   19 22 46 42 C9 3A 42 42
B158   B7 20 05 3E 08 CD 42 B1
B160   3D 32 42 42 C9 3A 42 42
B168   FE 3F 20 05 3E 37 CD 4A
B170   B1 3C 18 ED 3A 42 42 FE
B178   08 30 05 C6 08 CD 42 B1
B180   D6 08 18 DD 3A 42 42 FE
B188   38 38 05 D6 08 CD 4A B1
B190   C6 08 18 CD 3A 43 42 E6
```

```
B198  01 EE 01 32 43 42 C9 3A
B1A0  42 42 F5 E6 38 6F 26 00
B1A8  E5 D1 29 29 19 F1 E6 07
B1B0  5F 07 83 C6 86 5F 16 46
B1B8  3A 43 42 B7 28 01 13 19
B1C0  22 44 42 2A 46 42 CD DE
B1C8  B0 2A 48 42 CD 3A B1 2A
B1D0  44 42 36 7F CD 0A B1 FE
B1D8  08 20 07 CD 94 B1 C4 55
B1E0  B1 AF FE 09 20 07 CD 94
B1E8  B1 CC 65 B1 AF FE 0A 20
B1F0  04 CD 84 B1 AF FE 5B 20
B1F8  04 CD 74 B1 AF FE 3B 20
B200  08 2A 46 42 11 40 00 18
B208  0A FE 2D 20 0B 2A 46 42
B210  11 C0 FF 19 22 46 42 AF
B218  FE 1F CA 8E B2 CD E6 B2
B220  FE 5C 20 07 22 22 42 CD
B228  7C 35 AF FE 5D 20 26 ED
B230  5B 22 42 B7 ED 52 11 80
B238  00 19 7C B7 20 17 7D D6
B240  81 2A 22 42 77 ED 5B 48
B248  42 DF 23 D4 99 B4 CD 7C
B250  35 AF 32 43 42 FE 30 DA
B258  9F B1 FE 47 D2 9F B1 FE
B260  3A 38 05 FE 41 DA 9F B1
B268  D6 30 FE 0A 38 02 D6 07
B270  4F 3A 43 42 B7 20 12 CB
B278  01 CB 01 CB 01 CB 01 3E
B280  0F A6 B1 77 3E 09 C3 E2
B288  B1 3E F0 A6 18 F4 3A 42
B290  42 F5 E6 38 6F 26 00 E5
B298  D1 29 29 19 11 A0 46 F1
B2A0  E6 07 83 5F 19 22 44 42
B2A8  2A 46 42 CD DE B0 2A 44
B2B0  42 36 7F CD 0A B1 FE 08
B2B8  20 04 CD 55 B1 AF FE 09
B2C0  20 04 CD 65 B1 AF FE 0A
B2C8  20 04 CD 84 B1 AF FE 5B
B2D0  20 04 CD 74 B1 AF FE 1F
B2D8  CA 9F B1 B7 28 B0 CD E6
B2E0  B2 77 3E 09 18 D8 ED 5B
B2E8  42 42 16 00 2A 46 42 19
B2F0  22 48 42 C9 21 00 00 1A
B2F8  13 FE 20 28 FA 1B FE 0D
B300  37 C8 1A 13 D6 30 FE 0A
B308  38 02 D6 07 E6 0F 29 29
B310  29 29 85 6F 1A FE 20 C8
B318  FE 0D C8 18 E5 11 00 44
B320  1A 13 FE 0D CA A3 B3 FE
B328  20 28 F5 32 20 42 2A 48
```

11.03.94

```
B330  42 22 08 42 06 05 DD 21
B338  08 42 04 CD F4 B2 38 11
B340  05 DD 75 00 DD 74 01 DD
B348  23 DD 23 FE 0D 28 02 10
B350  E9 3E 06 90 32 07 42 CD
B358  87 A8 3A 20 42 C9 48 36
B360  7F CD 0A B1 FE 08 20 0A
B368  78 B9 28 06 04 36 20 2B
B370  18 ED FE 0D 20 02 77 C9
B378  FE 20 38 E3 FE 80 30 DF
B380  77 78 B7 28 DA 05 23 18
B388  D6 CD C9 01 ED 73 3C 42
B390  31 3C 42 FD E5 DD E5 D9
B398  08 F5 E5 D5 C5 D9 08 F5
B3A0  E5 D5 C5 31 88 42 11 28
B3A8  44 21 F2 AF DD 21 28 42
B3B0  06 0C C5 01 04 00 ED B0
B3B8  DD 7E 01 CD 03 A8 DD 7E
B3C0  00 CD 03 A8 DD 23 DD 23
B3C8  CD 84 A8 C1 10 E4 3E 46
B3D0  12 13 CD 87 A8 3A 2E 42
B3D8  4F 06 08 21 31 B0 3E 2D
B3E0  CB 01 30 01 7E 23 12 13
B3E8  10 F4 CD 84 A8 21 21 B0
B3F0  01 08 00 ED B0 2A 2C 42
B3F8  7E CD 03 A8 21 29 B0 01
B400  08 00 ED B0 2A 3C 42 7E
B408  CD 03 A8 23 7E CD 03 A8
B410  CD 6A B0 21 00 44 11 01
B418  44 01 27 00 36 20 ED B0
B420  2A 46 42 CD DE B0 21 00
B428  44 06 27 CD 5E B3 CD 1D
B430  B3 2A 08 42 ED 5B 0A 42
B438  01 A3 B3 C5 FE 41 CA AE
B440  B4 FE 44 CA 16 B5 FE 45
B448  CA 66 00 FE 46 CA 1B B7
B450  FE 47 CA 1B B5 FE 48 CA
B458  B4 B4 FE 49 CA E0 B7 FE
B460  4C CA 29 B6 FE 4D CA C5
B468  B6 FE 4F CA E7 B6 FE 50
B470  CA BC B7 FE 51 CA BE B4
B478  FE 52 CA 06 B5 FE 53 CA
B480  91 B5 FE 56 CA 29 B6 FE
B488  57 CA 47 B5 FE 58 CA 78
B490  B7 FE 5A CA F2 B4 C3 A3
B498  B3 D5 E5 22 48 42 7D E6
B4A0  3F 32 42 42 5F 7D 93 6F
B4A8  22 46 42 E1 D1 C9 CD 99
B4B0  B4 C3 8E B2 CD 99 B4 AF
B4B8  32 43 42 C3 9F B1 E5 06
B4C0  0B CD A4 AF E1 CD 0A B1
```

26.07.95 ..

```
/ B4C8  FE 0A 20 05 CD E9 AD 18
  B4D0  ED FE 5B 20 05 CD 2A B1
  B4D8  18 E4 FE 3B 20 05 2A 02
  B4E0  42 18 DB FE 2D 20 D7 06
  B4E8  0B C5 CD 2A B1 C1 10 F9
  B4F0  18 CC EB B7 ED 52 DA A3
  B4F8  B3 E5 C1 EB E5 D1 13 3A
  B500  0C 42 77 ED B0 C9 EB CD
  B508  99 B4 3A 0C 42 47 CD 6F
  B510  CF 13 24 10 F9 C9 7D CD
  B518  5B CE C9 22 3E 42 ED 78
  B520  3C 42 21 8C B3 E5 ED 73
  B528  3C 42 31 28 42 C1 D1 E1
  B530  F1 D9 08 C1 D1 E1 F1 D9
  B538  08 DD E1 FD E1 ED 7B 3C
  B540  42 E5 2A 3E 42 E3 C9 EB
  B548  3A 0C 42 47 CD 54 B5 13
  B550  24 10 F9 C9 7A FE 01 C2
  B558  7F CF 7B FE 68 DA 7F CF
  B560  FE 7A D2 7F CF C3 7B CF
  B568  CD 16 B1 21 39 B0 11 F0
  B570  44 01 09 00 ED B0 EB 06
  B578  0E CD 5E B3 21 F9 44 11
  B580  48 5B 7E FE 0D 28 05 12
  B588  13 23 18 F6 AF 12 36 20
  B590  C9 CD 68 B5 21 00 58 11
  B598  48 5B 01 00 00 CD 20 CE
  B5A0  2A 08 42 22 0E 42 CD EB
  B5A8  B5 3E 01 CD E5 B5 78 3C
  B5B0  3C CD E5 B5 7D CD E5 B5
  B5B8  7C CD E5 B5 7E CD E5 B5
  B5C0  23 10 F9 ED 5B 0A 42 DF
  B5C8  38 DC 3E 02 CD E5 B5 CD
  B5D0  E5 B5 3A 0C 42 CD E5 B5
  B5D8  3A 0D 42 CD E5 B5 11 48
  B5E0  5B CD 28 CE C9 11 48 5B
  B5E8  C3 1B 00 2A 0E 42 EB 2A
  B5F0  0A 42 B7 ED 52 7C B7 20
  B5F8  05 E5 C1 03 18 03 01 00
  B600  01 EB E5 09 22 0E 42 E1
  B608  41 C9 C5 D5 E5 2A 22 42
  B610  7E 23 F5 7C FE 59 20 09
  B618  11 48 5B CD 36 CE 21 00
  B620  58 F1 22 22 42 E1 D1 C1
  B628  C9 CD 68 B5 11 48 5B 21
  B630  00 58 01 00 00 22 22 42
  B638  CD 24 CE CD 36 CE 21 00
  B640  00 22 16 42 2B 22 14 42
  B648  CD 0A B6 FE 02 28 38 FE
  B650  01 C2 A3 B3 CD 0A B6 D6
  B658  02 47 CD 0A B6 6F CD 0A
```

```
B660  B6 67 ED 5B 16 42 DF 38
B668  03 22 16 42 ED 5B 14 42
B670  DF 30 03 22 14 42 CD 0A
B678  B6 4F 3A 20 42 FE 56 28
B680  01 71 23 10 DD 18 C1 CD
B688  0A B6 CD 0A B6 6F CD 0A
B690  B6 67 22 18 42 22 46 42
B698  11 48 5B CD 28 CE 21 42
B6A0  B0 11 40 45 01 11 00 ED
B6A8  B0 11 68 45 2A 14 42 CD
B6B0  42 A8 CD 84 A8 2A 16 42
B6B8  CD 42 A8 CD 84 A8 2A 18
B6C0  42 CD 42 A8 C9 EB B7 ED
B6C8  52 D8 E5 C1 03 2A 0C 42
B6D0  EB DF 38 04 ED B0 18 08
B6D8  09 EB 09 EB 1B 2B ED B8
B6E0  2A 0C 42 22 46 42 C9 DF
B6E8  D0 CD E9 AD 18 F9 3A 20
B6F0  42 FE 4F C0 D5 ED 5B 0E
B6F8  42 DF 30 1D ED 5B 0C 42
B700  DF 38 16 B7 ED 52 ED 5B
B708  10 42 19 EB 2A 00 42 7E
B710  FE ED 20 01 23 23 73 23
B718  72 D1 C9 3A 07 42 B7 28
B720  20 FE 01 28 16 21 0A 42
B728  47 05 11 1C 42 7E 23 23
B730  12 13 10 F9 3A 07 42 3D
B738  32 21 42 2A 08 42 22 1A
B740  42 ED 5B 1A 42 21 1C 42
B748  3A 21 42 47 1A BE 20 10
B750  D5 05 28 13 13 23 1A BE
B758  20 02 18 F5 D1 13 18 E5
B760  13 7B B2 C8 13 18 E5 D1
B768  ED 53 46 42 EB 23 22 1A
B770  42 2B 06 0B CD A4 AF C9
B778  CD 16 B1 21 53 B0 11 F0
B780  44 01 17 00 ED B0 11 18
B788  45 2A 08 42 CD 42 A8 E5
B790  C1 2A 0A 42 CD 87 A8 CD
B798  42 A8 CD 87 A8 E5 09 CD
B7A0  42 A8 CD 87 A8 E1 B7 ED
B7A8  42 CD 42 A8 CD 87 A8 C5
B7B0  E1 ED 4B 0A 42 B7 ED 42
B7B8  CD 42 A8 C9 06 01 CD A4
B7C0  AF CD 29 05 C0 E5 21 C8
B7C8  44 06 28 4E CD 0E 05 23
B7D0  10 F9 0E 0D CD 0E 05 E1
B7D8  ED 5B 0A 42 DF 38 DD C9
B7E0  7D CD 16 B1 E6 03 C6 30
B7E8  6F 26 00 22 EA 41 21 22
B7F0  49 22 E8 41 21 C8 44 22
```

```
B7F8  20 40 21 E8 41 C3 73 41
```

TAPE-VERSION DES MONITORS

```
7000  C3 96 7B F5 F5 0F 0F 0F
7008  0F CD 26 70 F1 CD 26 70
7010  F1 C9 FE A0 38 06 F5 3E
7018  30 12 13 F1 F5 CD 03 70
7020  3E 48 12 13 F1 C9 E6 0F
7028  C6 90 27 CE 40 27 12 13
7030  C9 F5 7C FE A0 38 05 3E
7038  30 12 13 7C CD 03 70 7D
7040  18 DB 7C CD 03 70 7D C3
7048  03 70 C5 F5 2A 00 42 7E
7050  FE ED 20 01 23 23 4E 23
7058  46 C5 E1 CD 31 70 CD 77
7060  7E F1 C1 C9 F5 3E 28 12
7068  13 F1 C9 F5 3E 29 18 F7
7070  F5 3E 2C 18 F2 F5 3E 41
7078  18 ED CD 75 70 18 F1 CD
7080  70 70 18 F1 CD 87 70 F5
7088  3E 20 18 DB F5 3E 2B 18
7090  D6 3E 02 18 02 3E 03 C3
7098  F0 70 F5 E6 38 0F 4F 06
70A0  00 F1 C9 C5 E5 F5 E6 07
70A8  4F 06 00 21 6C 76 09 7E
70B0  FE 4D 28 06 12 13 F1 E1
70B8  C1 C9 3A 05 42 B7 20 0F
70C0  CD 64 70 3E 02 CD F0 70
70C8  CD 6B 70 F1 E1 C1 C9 C6
70D0  03 CD 64 70 CD F0 70 CD
70D8  8C 70 2A 00 42 23 7E CD
70E0  12 70 CD 6B 70 AF 32 06
70E8  42 F1 E1 C1 04 04 23 C9
70F0  C5 E5 F5 E6 07 07 4F 06
70F8  00 21 74 76 09 7E 12 13
7100  23 7E 12 13 FE 4C 28 04
7108  F1 E1 C1 C9 3A 05 42 B7
7110  28 F6 1B 1B 3E 49 12 13
7118  3A 05 42 C6 57 12 13 F1
7120  E1 C1 F5 3A 06 42 B7 28
7128  05 AF 32 06 42 04 F1 C9
7130  C5 01 04 00 ED B0 CD 87
7138  70 C1 C9 F5 E6 38 0F 0F
7140  0F CD A3 70 F1 C9 C5 F5
```

7148	2A	00	42	23	7E	4F	06	00
7150	FE	80	38	02	06	FF	23	09
7158	CD	31	70	F1	C1	C9	F5	E6
7160	30	0F	0F	0F	0F	CD	F0	70
7168	F1	C9	CD	5E	71	F5	1B	1A
7170	FE	50	20	07	1B	CD	75	70
7178	3E	46	12	13	F1	C9	C5	CD
7180	9A	70	CB	09	21	80	76	09
7188	7E	12	23	13	7E	FE	20	28
7190	02	12	13	C1	C9	D6	20	CD
7198	7E	71	CD	70	70	C3	46	71
71A0	F5	E6	07	4F	0C	F1	0D	CA
71A8	7A	72	0D	CA	52	72	0D	28
71B0	51	0D	28	3A	0D	28	32	0D
71B8	28	24	0D	28	0C	21	DC	76
71C0	CD	9A	70	09	06	01	C3	30
71C8	71	E5	21	3C	77	06	02	CD
71D0	30	71	E1	CD	3B	71	CD	70
71D8	70	23	7E	C3	12	70	21	84
71E0	77	CD	30	71	06	01	C3	3B
71E8	71	21	80	77	18	F3	CB	5F
71F0	20	05	21	80	77	18	03	21
71F8	84	77	CD	30	71	06	01	C3
7200	5E	71	21	3C	77	CD	30	71
7208	FE	20	30	18	CB	5F	28	0E
7210	CD	7A	70	CD	64	70	06	01
7218	CD	5E	71	C3	6B	70	CD	13
7220	72	C3	7F	70	06	03	CB	5F
7228	28	18	FE	30	38	05	CD	7A
7230	70	18	06	CD	91	70	CD	70
7238	70	CD	64	70	CD	4A	70	C3
7240	6B	70	06	03	CD	39	72	CD
7248	70	70	FE	30	D2	75	70	C3
7250	91	70	CB	5F	28	13	21	90
7258	76	CD	30	71	06	01	F5	CD
7260	91	70	F1	CD	70	70	C3	5E
7268	71	21	3C	77	CD	30	71	06
7270	03	CD	5E	71	CD	70	70	C3
7278	4A	70	B7	20	06	21	98	77
7280	C3	E7	73	FE	08	20	18	21
7288	8C	77	CD	30	71	3E	30	CD
7290	6A	71	CD	70	70	CD	6A	71
7298	3E	27	12	13	06	01	C9	FE
72A0	10	20	0B	21	5C	77	CD	30
72A8	71	06	02	C3	46	71	21	60
72B0	77	CD	30	71	FE	18	28	F1
72B8	06	02	C3	95	71	FE	76	20
72C0	08	21	70	77	06	01	C3	30
72C8	71	21	3C	77	CD	30	71	06
72D0	01	CD	3B	71	CD	70	70	C3
72D8	A3	70	CD	9A	70	21	90	76

72E0	09	CD	30	71	F5	E6	38	FE
72E8	10	28	07	FE	20	30	03	CD
72F0	7A	70	06	01	F1	C3	A3	70
72F8	F5	E6	07	4F	0C	F1	0D	CA
7300	41	74	0D	CA	FA	73	0D	CA
7308	F1	73	0D	28	61	0D	28	4D
7310	0D	28	30	0D	28	0D	21	64
7318	77	CD	30	71	06	01	E6	38
7320	C3	12	70	CD	9A	70	E5	21
7328	90	76	09	CD	30	71	E6	38
7330	FE	10	28	07	FE	20	30	03
7338	CD	7A	70	06	02	E1	23	7E
7340	C3	12	70	CB	5F	28	0B	21
7348	54	77	CD	30	71	06	03	C3
7350	4A	70	21	58	77	CD	30	71
7358	06	01	C3	6A	71	21	54	77
7360	CD	30	71	CD	7E	71	CD	70
7368	70	06	03	C3	4A	70	FE	C3
7370	20	0B	21	50	77	CD	30	71
7378	06	03	C3	4A	70	FE	DB	20
7380	19	21	40	77	CD	30	71	CD
7388	7A	70	2A	00	42	23	CD	64
7390	70	7E	CD	12	70	06	02	C3
7398	6B	70	FE	D3	20	0C	21	44
73A0	77	CD	30	71	CD	8A	73	C3
73A8	7F	70	FE	E3	20	17	21	8C
73B0	77	CD	30	71	21	94	77	CD
73B8	30	71	1B	CD	70	70	06	01
73C0	3E	20	C3	5E	71	FE	EB	20
73C8	17	21	8C	77	CD	30	71	3E
73D0	01	CD	F0	70	CD	70	70	AF
73D8	32	05	42	06	01	C3	91	70
73E0	FE	F3	20	08	21	68	77	06
73E8	01	C3	30	71	21	6C	77	18
73F0	F6	21	50	77	CD	30	71	C3
73F8	63	73	CB	5F	28	38	FE	D9
7400	20	05	21	90	77	18	E0	FE
7408	C9	20	05	21	48	77	18	D7
7410	FE	E9	20	11	21	50	77	CD
7418	30	71	CD	64	70	06	01	CD
7420	91	70	C3	6B	70	21	3C	77
7428	CD	30	71	06	01	CD	95	70
7430	CD	70	70	C3	91	70	21	4C
7438	77	CD	30	71	06	01	C3	6A
7440	71	21	48	77	CD	30	71	06
7448	01	C3	7E	71	3A	05	42	B7
7450	28	01	23	23	7E	FE	40	38
7458	25	F5	E6	C0	07	07	07	07
7460	4F	06	00	21	CC	76	09	CD
7468	30	71	F1	F5	E6	38	0F	0F
7470	0F	C6	30	12	13	F1	CD	70

```
7478  70 06 02 C3 A3 70 CD 9A
7480  70 21 B0 76 09 CD 30 71
7488  06 02 C3 A3 70 23 7E FE
7490  C0 30 7A FE A0 D2 CE 75
7498  FE 80 30 71 FE 40 38 6D
74A0  F5 E6 07 4F F1 0C 0D CA
74A8  9C 75 0D CA 85 75 0D CA
74B0  B4 75 0D CA 67 75 0D CA
74B8  5B 75 0D CA 41 75 0D CA
74C0  1F 75 FE 70 30 47 FE 60
74C8  30 2B 21 3C 77 CD 30 71
74D0  CB 5F 28 11 CD 7A 70 06
74D8  49 FE 5F 20 02 06 52 78
74E0  12 13 06 02 C9 06 49 FE
74E8  4F 20 02 06 52 78 12 13
74F0  06 02 C3 7F 70 21 78 77
74F8  CB 5F 28 03 21 7C 77 CD
7500  30 71 06 02 CD 64 70 CD
7508  91 70 C3 6B 70 21 A4 77
7510  11 E8 41 CD 30 71 2A 00
7518  42 7E 06 01 C3 12 70 FE
7520  60 30 EA FE 4E 28 E6 21
7528  88 77 CD 30 71 06 30 FE
7530  46 28 08 06 31 FE 56 28
7538  02 06 32 78 12 13 06 02
7540  C9 FE 50 30 C8 21 48 77
7548  CD 30 71 1B 1B 06 49 FE
7550  4D 28 02 06 4E 78 12 13
7558  06 02 C9 FE 44 20 AE 21
7560  74 77 06 02 C3 30 71 21
7568  3C 77 CD 30 71 06 04 CB
7570  5F 20 09 CD 39 72 CD 70
7578  70 C3 5E 71 CD 5E 71 CD
7580  70 70 C3 39 72 FE 71 CA
7588  0D 75 21 44 77 CD 30 71
7590  21 9C 77 CD 30 71 1B 06
7598  02 C3 3B 71 FE 70 CA 0D
75A0  75 21 40 77 CD 30 71 06
75A8  02 CD 3B 71 21 A0 77 CD
75B0  30 71 1B C9 21 94 76 CB
75B8  5F 20 03 21 9C 76 CD 30
75C0  71 F5 CD 91 70 F1 CD 70
75C8  70 06 02 C3 5E 71 E6 07
75D0  FE 04 D2 0D 75 E6 03 07
75D8  07 07 07 4F 7E E6 18 0F
75E0  81 4F 06 00 21 FC 76 09
75E8  06 02 C3 30 71 22 00 42
75F0  C5 D5 11 E8 41 01 59 76
75F8  C5 AF 32 05 42 32 06 42
7600  7E FE 40 DA A0 71 FE 80
```

7608	DA	BD	72	FE	C0	DA	DA	72
7610	FE	CB	CA	4C	74	FE	ED	CA
7618	8D	74	FE	DD	20	22	3E	01
7620	32	05	42	32	06	42	23	22
7628	00	42	01	4A	76	C5	7E	FE
7630	DD	CA	49	76	FE	FD	CA	49
7638	76	FE	ED	CA	49	76	18	C0
7640	FE	FD	C2	F8	72	3E	02	18
7648	D7	C1	2A	00	42	2B	22	00
7650	42	3A	06	42	B7	C2	0D	75
7658	C9	AF	12	78	32	04	42	48
7660	06	00	2A	00	42	09	22	02
7668	42	D1	C1	C9	42	43	44	45
7670	48	4C	4D	41	42	43	44	45
7678	48	4C	53	50	49	58	49	59
7680	4E	5A	5A	20	4E	43	43	20
7688	50	4F	50	45	50	20	4D	20
7690	41	44	44	20	41	44	43	20
7698	53	55	42	20	53	42	43	20
76A0	41	4E	44	20	58	4F	52	20
76A8	4F	52	20	20	43	50	20	20
76B0	52	4C	43	20	52	52	43	20
76B8	52	4C	20	20	52	52	20	20
76C0	53	4C	41	20	53	52	41	20
76C8	53	4C	49	20	53	52	4C	20
76D0	42	49	54	20	52	45	53	20
76D8	53	45	54	20	52	4C	43	41
76E0	52	52	43	41	52	4C	41	20
76E8	52	52	41	20	44	41	41	20
76F0	43	50	4C	20	53	43	46	20
76F8	43	43	46	20	4C	44	49	20
7700	4C	44	44	20	4C	44	49	52
7708	4C	44	44	52	43	50	49	20
7710	43	50	44	20	43	50	49	52
7718	43	50	44	52	49	4E	49	20
7720	49	4E	44	20	49	4E	49	52
7728	49	4E	44	52	4F	55	54	49
7730	4F	55	54	44	4F	54	49	52
7738	4F	54	44	52	4C	44	20	20
7740	49	4E	20	20	4F	55	54	20
7748	52	45	54	20	50	4F	50	20
7750	4A	50	20	20	43	41	4C	4C
7758	50	55	53	48	44	4A	4E	5A
7760	4A	52	20	20	52	53	54	20
7768	44	49	20	20	45	49	20	20
7770	48	41	4C	54	4E	45	47	20
7778	52	52	44	20	52	4C	44	20
7780	49	4E	43	20	44	45	43	20
7788	49	4D	20	20	45	58	20	20
7790	45	58	58	20	28	53	50	29
7798	4E	4F	50	20	28	43	29	2C

```
77A0  2C 28 43 29 44 45 46 42
77A8  11 C8 44 CD ED 75 D5 C5
77B0  D5 E1 13 01 27 00 36 20
77B8  ED B0 C1 D1 2A 00 42 C5
77C0  D5 CD 42 70 CD 87 70 CD
77C8  87 70 3A 04 42 47 7E 23
77D0  CD 03 70 10 F9 D1 EB 01
77D8  10 00 09 EB D5 21 E8 41
77E0  7E B7 28 05 12 13 23 18
77E8  F7 E1 01 18 00 09 C1 EB
77F0  2A 02 42 10 B6 C9 42 43
77F8  3A 20 44 45 3A 20 48 4C
7800  3A 20 41 46 3A 20 42 43
7808  60 20 44 45 60 20 48 4C
7810  60 20 41 46 60 20 49 58
7818  3A 20 49 59 3A 20 53 50
7820  3A 20 50 43 3A 20 20 28
7828  48 4C 29 3A 20 20 20 28
7830  53 50 29 3A 20 53 5A 58
7838  48 58 50 4E 43 4E 41 4D
7840  45 20 20 20 20 53 54 41
7848  52 54 20 45 4E 44 20 20
7850  20 45 4E 54 52 59 54 41
7858  50 45 20 45 52 52 4F 52
7860  58 20 20 20 20 59 20 20
7868  20 20 58 2B 59 20 20 59
7870  2D 58 20 20 58 2D 59 20
7878  00 F0 01 28 00 11 01 F0
7880  36 03 ED B0 06 0C C5 06
7888  04 36 02 23 10 FB 06 06
7890  36 04 23 10 FB C1 10 EE
7898  06 02 36 02 23 10 FB 06
78A0  0C 36 09 23 10 FB 06 02
78A8  C5 06 06 36 02 23 10 FB
78B0  06 04 36 04 23 10 FB C1
78B8  10 EE E5 D1 13 36 05 01
78C0  BE 01 ED B0 06 08 C5 06
78C8  06 36 02 23 10 FB 06 04
78D0  36 09 23 36 09 23 23 36
78D8  01 23 36 01 23 23 10 F0
78E0  06 0A 36 04 23 10 FB C1
78E8  10 DC C9 C5 D5 E5 11 80
78F0  46 06 08 C5 CD 42 70 CD
78F8  84 70 E5 06 08 7E CD 03
7900  70 CD 87 70 23 10 F6 E1
7908  CD 84 70 01 08 00 ED B0
7910  C1 10 E0 E1 D1 C1 C9 CD
7918  2B 00 B7 28 FA FE 01 CA
7920  B0 7B C9 C5 D5 E5 21 C8
7928  44 11 C9 44 01 B7 01 36
7930  20 ED B0 E1 D1 C1 C9 E5
7938  D1 01 F8 FF 09 CD ED 75
```

```
7940  DF 38 FA 2A 00 42 C9 CD
7948  37 79 06 0B C3 A8 77 2A
7950  46 42 11 F8 FF 18 06 2A
7958  46 42 11 08 00 19 22 46
7960  42 C9 3A 42 42 B7 20 05
7968  3E 08 CD 4F 79 3D 32 42
7970  42 C9 3A 42 42 FE 3F 20
7978  05 3E 37 CD 57 79 3C 18
7980  ED 3A 42 42 FE 08 30 05
7988  C6 08 CD 4F 79 D6 08 18
7990  DD 3A 42 42 FE 38 38 05
7998  D6 08 CD 57 79 C6 08 18
79A0  CD 3A 43 42 E6 01 EE 01
79A8  32 43 42 C9 3A 42 42 F5
79B0  E6 38 6F 26 00 E5 D1 29
79B8  29 19 F1 E6 07 5F 07 83
79C0  C6 86 5F 16 46 3A 43 42
79C8  B7 28 01 13 19 22 44 42
79D0  2A 46 42 CD EB 78 2A 48
79D8  42 CD 47 79 2A 44 42 36
79E0  7F CD 17 79 FE 08 20 07
79E8  CD A1 79 C4 62 79 AF FE
79F0  09 20 07 CD A1 79 CC 72
79F8  79 AF FE 0A 20 04 CD 91
7A00  79 AF FE 5B 20 04 CD 81
7A08  79 AF FE 3B 20 08 2A 46
7A10  42 11 40 00 18 0A FE 2D
7A18  20 0B 2A 46 42 11 C0 FF
7A20  19 22 46 42 AF FE 1F CA
7A28  9B 7A CD F3 7A FE 5C 20
7A30  07 22 22 42 CD 7C 35 AF
7A38  FE 5D 20 26 ED 5B 22 42
7A40  B7 ED 52 11 80 00 19 7C
7A48  B7 20 17 7D D6 81 2A 22
7A50  42 77 ED 5B 48 42 DF 23
7A58  D4 97 7C CD 7C 35 AF 32
7A60  43 42 FE 30 DA AC 79 FE
7A68  47 D2 AC 79 FE 3A 38 05
7A70  FE 41 DA AC 79 D6 30 FE
7A78  0A 38 02 D6 07 4F 3A 43
7A80  42 B7 20 12 CB 01 CB 01
7A88  CB 01 CB 01 3E 0F A6 B1
7A90  77 3E 09 C3 EF 79 3E F0
7A98  A6 18 F4 3A 42 42 F5 E6
7AA0  38 6F 26 00 E5 D1 29 29
7AA8  19 11 A0 46 F1 E6 07 83
7AB0  5F 19 22 44 42 2A 46 42
7AB8  CD EB 78 2A 44 42 36 7F
7AC0  CD 17 79 FE 08 20 04 CD
7AC8  62 79 AF FE 09 20 04 CD
7AD0  72 79 AF FE 0A 20 04 CD
7AD8  91 79 AF FE 5B 20 04 CD
```

```
7AE0  81 79 AF FE 1F CA AC 79
7AE8  B7 28 B0 CD F3 7A 77 3E
7AF0  09 18 D8 ED 5B 42 42 16
7AF8  00 2A 46 42 19 22 48 42
7B00  C9 21 00 00 1A 13 FE 20
7B08  28 FA 1B FE 0D 37 C8 1A
7B10  13 D6 30 FE 0A 38 02 D6
7B18  07 E6 0F 29 29 29 29 85
7B20  6F 1A FE 20 C8 FE 0D C8
7B28  18 E5 11 00 44 1A 13 FE
7B30  0D CA B0 7B FE 20 28 F5
7B38  32 20 42 2A 48 42 22 08
7B40  42 06 05 DD 21 08 42 04
7B48  CD 01 7B 38 11 05 DD 75
7B50  00 DD 74 01 DD 23 DD 23
7B58  FE 0D 28 02 10 E9 3E 06
7B60  90 32 07 42 CD 87 70 3A
7B68  20 42 C9 48 36 7F CD 17
7B70  79 FE 08 20 0A 78 B9 28
7B78  F3 04 36 20 2B 18 ED FE
7B80  0D 20 02 77 C9 FE 20 38
7B88  E3 FE 80 30 DF 77 78 B7
7B90  28 DA 05 23 18 D6 CD C9
7B98  01 ED 73 3C 42 31 3C 42
7BA0  FD E5 DD E5 D9 08 F5 E5
7BA8  D5 C5 D9 08 F5 E5 D5 C5
7BB0  31 88 42 11 28 44 21 F6
7BB8  77 DD 21 28 42 06 0C C5
7BC0  01 04 00 ED B0 DD 7E 01
7BC8  CD 03 70 DD 7E 00 CD 03
7BD0  70 DD 23 DD 23 CD 84 70
7BD8  C1 10 E4 3E 46 12 13 CD
7BE0  87 70 3A 2E 42 4F 06 08
7BE8  21 35 78 3E 2D CB 01 30
7BF0  01 7E 23 12 13 10 F4 CD
7BF8  84 70 21 25 78 01 08 00
7C00  ED B0 2A 2C 42 7E CD 03
7C08  70 21 2D 78 01 08 00 ED
7C10  B0 2A 3C 42 7E CD 03 70
7C18  23 7E CD 03 70 CD 77 78
7C20  21 00 44 11 01 44 01 27
7C28  00 36 20 ED B0 2A 46 42
7C30  CD EB 78 21 00 44 06 27
7C38  CD 6B 7B CD 2A 7B 2A 08
7C40  42 ED 5B 0A 42 01 B0 7B
7C48  C5 FE 41 CA AC 7C FE 44
7C50  CA BC 7C FE 45 CA 66 00
7C58  FE 46 CA A4 7E FE 47 CA
7C60  25 7D FE 48 CA B2 7C FE
7C68  4C CA 51 7D FE 4D CA 4E
7C70  7E FE 4F CA 70 7E FE 50
```

7C78	CA	45	7F	FE	51	CA	BC	7C
7C80	FE	53	CA	69	7F	FE	56	CA
7C88	51	7D	FE	58	CA	01	7F	FE
7C90	5A	CA	F0	7C	C3	B0	7B	D5
7C98	E5	22	48	42	7D	E6	3F	32
7CA0	42	42	5F	7D	93	6F	22	46
7CA8	42	E1	D1	C9	CD	97	7C	C3
7CB0	9B	7A	CD	97	7C	AF	32	43
7CB8	42	C3	AC	79	E5	06	0B	CD
7CC0	A8	77	E1	CD	17	79	FE	0A
7CC8	20	05	CD	ED	75	18	ED	FE
7CD0	5B	20	05	CD	37	79	18	E4
7CD8	FE	3B	20	05	2A	02	42	18
7CE0	DB	FE	2D	20	D7	06	0B	C5
7CE8	CD	37	79	C1	10	F9	18	CC
7CF0	EB	B7	ED	52	DA	B0	7B	E5
7CF8	C1	EB	E5	D1	13	3A	0C	42
7D00	77	ED	B0	C9	CD	ED	01	32
7D08	27	44	C9	CD	1F	02	32	27
7D10	44	C9	CD	04	7D	F5	3A	26
7D18	44	EE	0A	32	26	44	F1	C9
7D20	CD	0B	7D	18	F0	22	3E	42
7D28	ED	7B	3C	42	21	99	7B	E5
7D30	ED	73	3C	42	31	28	42	C1
7D38	D1	E1	F1	D9	08	C1	D1	E1
7D40	F1	D9	08	DD	E1	FD	E1	ED
7D48	7B	3C	42	E5	2A	3E	42	E3
7D50	C9	CD	23	79	21	3D	78	11
7D58	F0	44	01	19	00	ED	B0	21
7D60	00	00	22	16	42	2B	22	14
7D68	42	CD	4C	02	21	2A	2A	22
7D70	26	44	CD	04	7D	FE	55	C2
7D78	CC	7D	21	18	45	06	06	CD
7D80	04	7D	77	23	10	F9	CD	04
7D88	7D	FE	78	28	4D	FE	3C	C2
7D90	CC	7D	CD	04	7D	47	CD	04
7D98	7D	6F	CD	04	7D	67	85	4F
7DA0	ED	5B	14	42	DF	30	03	22
7DA8	14	42	ED	5B	16	42	DF	38
7DB0	03	22	16	42	CD	04	7D	5F
7DB8	3A	20	42	FE	56	28	01	73
7DC0	23	7B	81	4F	10	E4	CD	12
7DC8	7D	B9	28	BA	21	56	78	11
7DD0	68	45	01	0A	00	ED	B0	C3
7DD8	B0	7B	CD	04	7D	6F	CD	04
7DE0	7D	67	22	18	42	11	20	45
7DE8	2A	14	42	CD	42	70	CD	84
7DF0	70	2A	16	42	CD	42	70	CD
7DF8	84	70	2A	18	42	CD	42	70
7E00	CD	97	7C	C3	B0	7B	CD	23
7E08	79	21	3D	78	11	F0	44	01

```
7E10   05 00 ED B0 EB 06 06 CD
7E18   6B 7B 21 F5 44 11 E8 41
7E20   7E FE 0D 28 05 12 13 23
7E28   18 F6 AF 12 36 20 C9 2A
7E30   0E 42 EB 2A 0A 42 B7 ED
7E38   52 7C B7 20 05 E5 C1 03
7E40   18 03 01 00 01 EB E5 09
7E48   22 0E 42 E1 41 C9 EB B7
7E50   ED 52 D8 E5 C1 03 2A 0C
7E58   42 EB DF 38 04 ED B0 18
7E60   08 09 EB 09 EB 1B 2B ED
7E68   B8 2A 0C 42 22 46 42 C9
7E70   DF D0 CD ED 75 18 F9 3A
7E78   20 42 FE 4F C0 D5 ED 5B
7E80   0E 42 DF 30 1D ED 5B 0C
7E88   42 DF 38 16 B7 ED 52 ED
7E90   5B 10 42 19 EB 2A 00 42
7E98   7E FE ED 20 01 23 23 73
7EA0   23 72 D1 C9 3A 07 42 B7
7EA8   28 20 FE 01 28 16 21 0A
7EB0   42 47 05 11 1C 42 7E 23
7EB8   23 12 13 10 F9 3A 07 42
7EC0   3D 32 21 42 2A 08 42 22
7EC8   1A 42 ED 5B 1A 42 21 1C
7ED0   42 3A 21 42 47 1A BE 20
7ED8   10 D5 05 28 13 13 23 1A
7EE0   BE 20 02 18 F5 D1 13 18
7EE8   E5 13 7B B2 C8 13 18 E5
7EF0   D1 ED 53 46 42 EB 23 22
7EF8   1A 42 2B 06 0B CD A8 77
7F00   C9 CD 23 79 21 60 78 11
7F08   F0 44 01 17 00 ED B0 11
7F10   18 45 2A 08 42 CD 42 70
7F18   E5 C1 2A 0A 42 CD 87 70
7F20   CD 42 70 CD 87 70 E5 09
7F28   CD 42 70 CD 87 70 E1 B7
7F30   ED 42 CD 42 70 CD 87 70
7F38   C5 E1 ED 4B 0A 42 B7 ED
7F40   42 CD 42 70 C9 06 01 CD
7F48   A8 77 CD 29 05 C0 E5 21
7F50   C8 44 06 28 4E CD 0E 05
7F58   23 10 F9 0E 0D CD 0E 05
7F60   E1 ED 5B 0A 42 DF 38 DD
7F68   C9 22 0E 42 CD 06 7E 21
7F70   2A 2A 22 26 44 CD 3F 02
7F78   3E 55 CD 0B 7D 21 E8 41
7F80   06 06 7E FE 20 30 02 3E
7F88   20 CD 0B 7D 23 10 F3 CD
7F90   2F 7E 3E 3C CD 0B 7D 78
7F98   CD 0B 7D 7D CD 0B 7D 7C
7FA0   CD 0B 7D 85 4F 7E CD 0B
```

7FA8	7D	81	4F	23	10	F7	CD	20
7FB0	7D	ED	5B	0A	42	DF	38	D7
7FB8	3E	78	CD	0B	7D	2A	0C	42
7FC0	7D	CD	0B	7D	7C	CD	0B	7D
7FC8	C3	B0	7B	FF	FF	FF	FF	FF
7FD0	FF	FF	FF	FF	FF	FF	FF	FF
7FD8	FF	FF	FF	FF	FF	FF	FF	FF
7FE0	FF	FF	FF	FF	FF	FF	FF	FF
7FE8	FF	FF	FF	FF	FF	FF	FF	FF
7FF0	FF	FF	FF	FF	FF	FF	FF	FF
7FF8	FF	FF	FF	FF	FF	FF	FF	FF

Ein Hilfsprogramm fuer Kassettenbenutzer

TAPE-EDITOR

```
Loading TAPEDI                              *P
LENGTH OF PROGRAM 09C9. ENTRY 50FB.
SYSTEM-TAPE COPY & EDITOR   (c) 1983
written by Juergen Buchmueller
L-LOAD  S-SAVE  M-MERGE E-EDIT  A-SCHUTZ
B-BASIC N-NAME  C-CHANGE ENTRY
Command ?
```

```
4800   03020425954455320465 24F4D204144
4810   524553532030303030 2E20434845434B
4820   53554D20303020204C454E475448204F
4830   462050524F4752414D20303030302E20
4840   454E545259203030302E2020202020 20
4850   4C6F6164696E6720536176696E672020
4860   4D6572676520746F204F6C64204E616D
4870   65202E2E2E2E2E2E204E6577204E616D
80 BYTES FROM ADRESS 4800. CHECKSUM F6
4800   00 BYTES FROM ADRESS 0000. CHECK
4820   SUM 00  LENGTH OF PROGRAM 0000.
4840   ENTRY 0000.      Loading Saving
4860   Merge to Old Name ...... New Nam
```

An den obigen Bildschirmausdrucken konnen Sie schon ungefahr erkennen, was Ihnen der "Tape-Editor" bietet.
Sie konnen mit ihm SYSTEM-Bander laden, abspeichern und mehrere SYSTEM-Bander zu einem Programm zusammenfassen.
Der Name und die Startadresse des zu schreibenden Bandes konnen geandert werden.
Desweiteren konnen die Programme vor dem Abspeichern in hexadezimal oder ASCII editiert werden, wobei Sie sogar Blocke loschen oder einfugen konnen.
Ferner bietet der Tape-Editor die Moglichkeit, Bander geschutzt aufzuzeichnen, d.h. das Programm wird mit einem AUTO-Start versehen, der Video-DCB wird 'verbogen' (damit lasst sich das Band mit einem Monitor nicht mehr laden) und die RST-Tasten werden desaktiviert.
Alles in allem ein sehr nutzliches Programm fur Colour-Genie-Besitzer, die sich eine Floppydiskstation noch nicht leisten konnen !

Die Bedienung des Tape-Editors

Das Menue bietet folgende Befehle:

(L)oad - Ladt ein Maschinenprogramm von Kassette in den Speicher. Dabei wird nicht auf die Originaladdressen geladen, sondern in einen Buffer. So lassen sich die Programme auch andern (siehe EDIT), da die Prufsummenbytes ebenfalls im Buffer stehen.

(S)ave - Speichert ein im Buffer stehendes Programm wieder auf Kassette. Dieser Vorgang kann beliebig oft wiederholt werden.

(M)erge - Hangt an das im Buffer stehende Proogramm ein weiteres von Kassette an. Dabei wird als Startadresse fur das komplette Programm die des zuletzt angehangten angenommen. Der Name ist der des zuerst geladenen Programmes.

(E)dit - Dient zum Verandern eines im Buffer stehenden Programmes. Es wird jeweils ein Block mit zugehoriger Lange und Prufsumme angezeigt.
Der Edit Befehl hat folgende Sub-Kommandos:

<CLEAR> dient zum umschalten zwischen Hex-Edit und ASCII-Edit.

<BREAK> dient zum verlassen des Edit-Befehls.

<F1> loscht den augenblicklich angezeigten Block aus dem Buffer.

<F2> fugt einen Block vor den augenblicklich angezeigten ein. Die Adresse und und Lange wird gesondert abgefragt und muß in der Form XXXX,YY eingegeben werde.

<F3> sucht das Byte im Buffer, das auf die angegebene Adresse bei normalem SYSTEM-Befehl geladen wurde. Die Adresse wird gesondert abgefragt.

<RETURN> schaltet einen Block vorwarts.

Die Prufsumme wird bei jeder Andderunng fur den jeweiligen Block neu berechnet und im Buffer abgelegt.

(A) Schutz Das im Buffer stehende Programm wird gegen unerlaubtes Kopieren geschutzt, kann aber mit dem TAPE-EDITOR trotzdem wieder geladen werden.

(B)asic - Basic-Warmstart.

(N)ame - dient zum andern des Programmnamens eines im Buffer befindlichen Programms.

(C)hange Entry - dient zum andern der Startadresse eines im Buffer befindlichen Programms. Die Adresse muß vierstellig eingegeben werden.

Zu Beginn des Ladevorgangs wird der Name des gefundenen Programms angezeigt, nach erfolgtem Laden die Lange des Programms und dessen Startadresse.

Bei einer Fehlerhaften Prufsumme, oder einem falschen Blockheader wird die entsprechende Fehlermeldung angezeigt.

Das Programm beginnt bei Adresse 4800H und endet bei Adresse 5157H. Die Startadresse ist 50FBH.

Das Listing beginnt auf der folgenden Seite.

Das Hex-Listing

4800	30	30	20	42	59	54	45	53
4808	20	46	52	4F	4D	20	41	44
4810	52	45	53	53	20	30	30	30
4818	30	2E	20	43	48	45	43	4B
4820	53	55	4D	20	30	30	20	20
4828	4C	45	4E	47	54	48	20	4F
4830	46	20	50	52	4F	47	52	41
4838	4D	20	30	30	30	30	2E	20
4840	45	4E	54	52	59	20	30	30
4848	30	30	2E	20	20	20	20	20
4850	4C	6F	61	64	69	6E	67	20
4858	53	61	76	69	6E	67	20	20
4860	4D	65	72	67	65	20	74	6F
4868	20	4F	6C	64	20	4E	61	6D
4870	65	20	2E	2E	2E	2E	2E	2E
4878	20	4E	65	77	20	4E	61	6D
4880	65	20	3F	20	49	6E	73	65
4888	72	74	20	41	64	72	65	73
4890	73	2C	42	79	74	65	73	20
4898	4F	6C	64	20	45	6E	74	72
48A0	79	20	30	30	30	30	20	4E
48A8	65	77	20	45	6E	74	72	79
48B0	20	3F	20	46	69	6E	64	20
48B8	50	72	6F	67	72	61	6D	2D
48C0	41	64	72	65	73	73	20	20
48C8	3C	02	16	40	00	00	56	00
48D0	3C	02	16	40	E3	03	3C	00
48D8	3C	03	E2	41	C3	00	00	00
48E0	3C	03	BE	41	C3	00	00	00
48E8	43	68	65	63	6B	73	75	6D
48F0	20	45	72	72	6F	72	0D	00
48F8	4E	6F	20	53	59	53	54	45
4900	4D	20	54	61	70	65	0D	00
4908	42	75	66	66	65	72	20	66
4910	75	6C	6C	0D	00	4E	6F	20
4918	50	72	6F	67	72	61	6D	20
4920	69	6E	20	42	75	66	66	65
4928	72	0D	00	53	59	53	54	45
4930	4D	2D	54	41	50	45	20	43
4938	4F	50	59	20	26	20	45	44
4940	49	54	4F	52	20	20	20	28
4948	63	29	20	31	39	38	33	20
4950	20	20	20	77	72	69	74	74
4958	65	6E	20	62	79	20	4A	75
4960	65	72	67	65	6E	20	42	75
4968	63	68	6D	75	65	6C	6C	65
4970	72	20	20	20	20	20	20	20
4978	20	20	20	20	20	20	20	20
4980	20	20	20	20	20	20	20	20
4988	20	20	20	20	20	20	20	20
4990	20	20	20	20	20	20	20	20

```
4998   20 20 20 20 20 20 20 20
49A0   20 20 20 4C 2D 4C 4F 41
49A8   44 20 20 53 2D 53 41 56
49B0   45 20 20 4D 2D 4D 45 52
49B8   47 45 20 45 2D 45 44 49
49C0   54 20 20 41 2D 53 43 48
49C8   55 54 5A 42 2D 42 41 53
49D0   49 43 20 4E 2D 4E 41 4D
49D8   45 20 20 43 2D 43 48 41
49E0   4E 47 45 20 45 4E 54 52
49E8   59 20 20 20 20 20 20 20
49F0   20 20 20 43 6F 6D 6D 61
49F8   6E 64 20 3F 20 00 00 00
4A00   00 00 00 00 00 00 00 00
4A08   00 00 00 F5 0F 0F 0F 0F
4A10   CD 14 4A F1 E6 0F C6 30
4A18   FE 3A 38 02 C6 07 12 13
4A20   C9 7C CD 0B 4A 7D CD 0B
4A28   4A C9 21 00 F0 06 10 C5
4A30   06 06 36 02 23 10 FB 06
4A38   08 36 08 23 36 08 23 36
4A40   0B 23 36 0B 23 10 F2 23
4A48   23 C1 10 E3 06 28 36 04
4A50   23 10 FB 06 08 C5 06 06
4A58   36 02 23 10 FB 06 22 36
4A60   05 23 10 FB C1 10 EE C9
4A68   21 00 44 11 01 44 01 FF
4A70   03 36 20 ED B0 21 00 44
4A78   22 20 40 C9 01 07 00 DD
4A80   09 18 0C DD 21 5F 51 DD
4A88   7E 00 FE 78 CA 05 51 DD
4A90   7E 00 FE 3C 28 3E FE 55
4A98   28 E2 FE 78 28 E5 3E 3C
4AA0   DD 77 00 DD 46 01 DD 6E
4AA8   02 DD 66 03 11 00 48 78
4AB0   CD 0B 4A 11 15 48 CD 21
4AB8   4A 7C 85 4F DD E5 E1 23
4AC0   23 23 23 7E 81 4F 23 10
4AC8   FA 77 11 24 48 CD 0B 4A
4AD0   CD 32 4B C9 DD 46 01 DD
4AD8   6E 02 DD 66 03 11 00 44
4AE0   DD E5 0E 10 CD 21 4A 13
4AE8   13 DD 7E 04 CD 0B 4A 23
4AF0   05 28 09 0D DD 23 20 F1
4AF8   13 13 18 E6 DD E1 CD A3
4B00   4A DD 46 01 DD 6E 02 DD
4B08   66 03 DD E5 0E 20 CD 21
4B10   4A 13 13 DD 7E 04 FE 20
4B18   38 04 FE 80 38 02 3E 2E
4B20   12 13 23 05 28 09 0D DD
4B28   23 20 E8 13 13 18 DD DD
4B30   E1 C9 11 80 46 21 00 48
4B38   01 28 00 ED B0 C9 CD ED
```

```
4B40  01 77 BE 20 0F 23 32 27
4B48  44 C9 21 E8 48 18 0D 21
4B50  F8 48 18 08 21 08 49 18
4B58  03 21 15 49 11 28 44 ED
4B60  53 20 40 CD A7 28 CD 82
4B68  4B C3 08 51 CD 68 4A 21
4B70  00 F0 11 01 F0 01 FF 03
4B78  36 03 ED B0 3E 0F CD 33
4B80  00 C9 21 58 51 11 59 51
4B88  01 A7 6E 36 00 ED B0 21
4B90  00 00 11 3A 48 CD 21 4A
4B98  11 46 48 CD 21 4A C9 CD
4BA0  82 4B CD 6C 4B 21 50 48
4BA8  11 00 44 01 08 00 ED B0
4BB0  CD 4C 02 21 58 51 CD 3E
4BB8  4B FE 55 20 92 11 08 44
4BC0  06 06 CD 3E 4B 12 13 10
4BC8  F9 CD 3E 4B FE 78 28 1D
4BD0  CD 3E 4B 47 CD 3E 4B 4F
4BD8  CD 3E 4B 81 4F CD 3E 4B
4BE0  81 4F 10 F9 CD 3E 4B B9
4BE8  28 DF C3 4A 4B CD 3E 4B
4BF0  5F CD 3E 4B 57 ED 53 FD
4BF8  49 22 FF 49 11 58 51 B7
4C00  ED 52 22 01 4A 11 3A 48
4C08  CD 21 4A 2A FD 49 11 46
4C10  48 CD 21 4A C3 08 51 3A
4C18  58 51 FE 55 C2 59 4B CD
4C20  6C 4B 21 58 48 11 00 44
4C28  01 08 00 ED B0 21 59 51
4C30  01 06 00 ED B0 21 2A 2A
4C38  22 26 44 CD 3F 02 21 58
4C40  51 ED 4B 01 4A 7E 32 27
4C48  44 CD 1F 02 23 0B 78 B1
4C50  20 F3 C3 08 51 3A 58 51
4C58  FE 55 C2 59 4B CD 6C 4B
4C60  21 60 48 11 00 44 01 09
4C68  00 ED B0 21 59 51 01 06
4C70  00 ED B0 3E 2D 12 13 CD
4C78  4C 02 CD ED 01 FE 55 C2
4C80  4F 4B 06 06 CD ED 01 12
4C88  13 10 F9 2A FF 49 2B 2B
4C90  2B C3 C9 4B 29 29 29 E5
4C98  29 29 D1 19 C9 3A 58 51
4CA0  FE 55 C2 59 4B DD 21 58
4CA8  51 21 00 00 22 03 4A 22
4CB0  05 4A CD 6C 4B CD 2A 4A
4CB8  CD 8F 4A AF 32 04 4A CD
4CC0  A3 4A 3A 07 4A B7 C2 0E
4CC8  4E 3A 03 4A E6 0F 4F 3A
4CD0  03 4A E6 F0 0F 0F 0F 0F
4CD8  6F 26 00 CD 94 4C CB 01
4CE0  06 44 09 01 06 00 09 3A
4CE8  04 4A B7 28 01 23 22 05
4CF0  4A 4E 36 CA CD 49 00 71
4CF8  FE 08 20 17 3A 03 4A B7
4D00  28 07 3D 32 03 4A C3 BB
4D08  4C DD 7E 01 3D 32 03 4A
4D10  C3 BB 4C FE 09 20 10 3A
4D18  03 4A 3C DD BE 01 20 01
4D20  AF 32 03 4A C3 BB 4C FE
```

```
4D28  5B 20 1A 3A 03 4A D6 10
4D30  30 0D C6 10 E6 0F 4F DD
4D38  7E 01 D6 10 E6 F0 B1 32
4D40  03 4A C3 BB 4C FE 0A 20
4D48  1F DD 7E 01 B7 3A 03 4A
4D50  28 0E 3A 03 4A C6 10 DD
4D58  BE 01 38 06 E6 0F D6 10
4D60  C6 10 32 03 4A C3 BB 4C
4D68  FE 1F 20 0B 3A 07 4A EE
4D70  01 32 07 4A C3 BB 4C FE
4D78  0D 20 26 CD 6C 4B CD 2A
4D80  4A DD 4E 01 79 B7 20 04
4D88  06 01 18 02 06 00 03 03
4D90  03 03 03 DD 09 CD 8F 4A
4D98  21 00 00 22 03 4A C3 BF
4DA0  4C FE 01 CA 05 51 FE 5C
4DA8  CA A4 4F FE 5D CA 06 50
4DB0  FE 5E CA 84 50 FE 30 DA
4DB8  BF 4C FE 47 D2 BF 4C FE
4DC0  41 30 07 FE 3A 38 03 C3
4DC8  BF 4C 77 D6 30 FE 0A 38
4DD0  02 D6 07 4F 3A 04 4A B7
4DD8  28 0B 3E F0 CD FD 4D CD
4DE0  8F 4A C3 17 4D CB 09 CB
4DE8  09 CB 09 CB 09 3E 0F CD
4DF0  FD 4D 3E 01 32 04 4A CD
4DF8  8F 4A C3 BF 4C 2A 03 4A
4E00  26 00 DD E5 D1 13 13 13
4E08  13 19 A6 B1 77 C9 3A 03
4E10  4A E6 1F 4F 3A 03 4A E6
4E18  E0 07 07 07 6F 26 00 CD
4E20  94 4C 06 44 09 01 AE 02
4E28  09 22 05 4A 4E 36 CA CD
4E30  49 00 71 FE 08 CA FC 4C
4E38  FE 09 CA 17 4D FE 5B CA
4E40  2B 4D FE 0A CA 49 4D FE
4E48  1F CA 6C 4D FE 0D CA 7B
4E50  4D FE 01 CA 05 51 FE 5C
4E58  CA A4 4F FE 5D CA 06 50
4E60  FE 5E CA 84 50 4F 77 2A
4E68  03 4A 26 00 DD E5 D1 13
4E70  13 13 13 19 71 CD 8F 4A
4E78  C3 17 4D 3A 58 51 FE 55
4E80  C2 59 4B 21 FF BF ED 5B
4E88  FF 49 B7 ED 52 11 28 00
4E90  ED 52 DA 54 4B 21 F8 BF
4E98  11 5E 51 B7 ED 52 E5 C1
4EA0  21 F8 BF 11 FF BF ED B8
4EA8  11 5F 51 21 C8 48 01 07
4EB0  00 ED B0 2A FF 49 11 04
4EB8  00 19 E5 E5 D1 13 1A 32
4EC0  DD 48 32 E5 48 13 1A 32
4EC8  DE 48 32 E6 48 4F 1B 1A
4ED0  81 4F C6 E6 32 DF 48 79
4ED8  C6 C2 32 E7 48 2A DD 48
4EE0  22 FD 49 D1 21 D0 48 01
4EE8  07 00 ED B0 21 D8 48 01
4EF0  10 00 ED B0 EB 36 78 23
4EF8  36 00 23 36 00 23 22 FF
4F00  49 11 58 51 B7 ED 52 22
```

4F08	01	4A	C3	05	4C	3A	58	51
4F10	FE	55	C2	59	4B	CD	6C	4B
4F18	11	00	44	21	69	48	01	1B
4F20	00	ED	B0	ED	53	20	40	11
4F28	09	44	21	59	51	01	06	00
4F30	ED	B0	21	00	42	06	06	E5
4F38	C5	36	20	23	10	FB	C1	E1
4F40	CD	40	00	11	59	51	06	06
4F48	7E	FE	20	30	02	3E	20	12
4F50	23	13	10	F4	C3	08	51	3A
4F58	58	51	FE	55	C2	59	4B	CD
4F60	6C	4B	2A	FF	49	2B	56	2B
4F68	5E	EB	11	A2	48	CD	21	4A
4F70	11	00	44	21	98	48	01	1A
4F78	00	ED	B0	ED	53	20	40	21
4F80	00	42	06	04	CD	40	00	06
4F88	04	CD	F2	50	2A	FF	49	2B
4F90	72	2B	73	EB	C3	0E	4C	7E
4F98	23	D6	30	FE	0A	38	02	D6
4FA0	07	E6	0F	C9	DD	4E	01	06
4FA8	00	79	B7	20	01	04	03	03
4FB0	03	03	03	DD	E5	E1	22	09
4FB8	4A	09	22	05	4A	2A	FF	49
4FC0	AF	ED	42	22	FF	49	21	FF
4FC8	BF	AF	ED	4B	05	4A	ED	42
4FD0	E5	C1	2A	05	4A	ED	5B	09
4FD8	4A	ED	B0	2A	FF	49	11	58
4FE0	51	AF	ED	52	22	01	4A	11
4FE8	3A	48	CD	21	4A	DD	7E	00
4FF0	FE	78	28	12	CD	6C	4B	CD
4FF8	2A	4A	21	00	00	22	03	4A
5000	CD	8F	4A	C3	BF	4C	CD	6C
5008	4B	21	84	48	11	00	44	01
5010	14	00	ED	B0	ED	53	20	40
5018	21	00	42	06	07	CD	40	00
5020	CD	F2	50	ED	53	05	4A	23
5028	CD	E5	50	32	08	4A	4F	06
5030	00	B7	20	01	04	03	03	03
5038	03	03	21	FF	BF	AF	ED	42
5040	22	09	4A	ED	5B	FF	49	DF
5048	DA	54	4B	2A	FF	49	09	22
5050	FF	49	2A	09	4A	DD	E5	C1
5058	ED	42	E5	C1	03	11	FF	BF
5060	2A	09	4A	ED	B8	2A	05	4A
5068	3A	08	4A	DD	77	01	DD	75
5070	02	DD	74	03	DD	E5	E1	23
5078	23	23	23	47	36	00	23	10
5080	FB	C3	DB	4F	CD	6C	4B	11
5088	00	44	21	B3	48	01	14	00
5090	ED	B0	ED	53	20	40	21	00
5098	42	06	04	CD	40	00	CD	F2
50A0	50	FD	21	5F	51	FD	7E	00
50A8	FE	78	28	2D	FD	46	01	FD
50B0	6E	02	FD	66	03	DF	28	16
50B8	23	10	FA	FD	4E	01	06	00

50C0	79	B7	20	01	04	03	03	03
50C8	03	03	FD	09	18	D7	FD	7E
50D0	01	90	32	03	4A	FD	E5	DD
50D8	E1	CD	6C	4B	CD	2A	4A	CD
50E0	8F	4A	C3	BF	4C	CD	97	4F
50E8	0F	0F	0F	0F	4F	CD	97	4F
50F0	81	C9	CD	E5	50	57	CD	E5
50F8	50	5F	C9	31	88	42	AF	32
5100	23	40	CD	82	4B	CD	6C	4B
5108	CD	7C	35	21	28	48	11	78
5110	44	01	28	00	ED	B0	21	2B
5118	49	11	C8	44	01	D2	00	ED
5120	B0	ED	53	20	40	3E	0E	CD
5128	33	00	CD	49	00	FE	45	CA
5130	9D	4C	FE	4C	CA	9F	4B	FE
5138	53	CA	17	4C	FE	4D	CA	55
5140	4C	FE	42	CA	66	00	FE	41
5148	CA	7B	4E	FE	4E	CA	0D	4F
5150	FE	43	CA	57	4F	C3	2A	51

Der Aufbau des Inhaltsverzeichnisses einer Diskette

Da im folgenden einige Programme gelistet werden, die sich mit dem Inhaltsverzeichnis von Disketten beschaftigen, wollen wir Ihnen nun den Aufbau des Inhaltsverzeichnisses etwas genauer beschreiben. Wenn Sie dieses Kapitel gelesen und verstanden haben, sind Sie in der Lage, zum Beispiel versehentlich geloschte Files auf einer Diskette wieder lesbar zu machen und vieles andere mehr:

Das Inhaltsverzeichnis entspricht praktisch einem normalen Programm und hat also auch einen eigenen Eintrag im Inhaltsverzeichnis, namlich DIR/SYS.
Dieser File ist 15 Sektoren lang, die folgende Bedeutung haben:

Der erste Sektor enthalt die sogenannte DBT (= Disketten Belegungs-Tabelle).
Die ersten 192 Bytes dieses Sektors entsprechen jeweils einem "Lump", einer Zusammenfassung von mehreren "Granules" (= 5 Sektoren).
Wieviele Granules zu einem Lump zusammengefaßt werden, hangt von der Speicherkapazitat der Diskette ab, da maximal 192 Lumps verwaltet werden konnen.
Nach dem Einschalten ist Ihr Colour-Genie auf 3 Granules pro Lump eingestellt.
Jedes Bit eines Bytes der DBT entspricht nun einem Granule des zugehorigen Lumps. Ist das Bit 1, ist das Granule belegt, ist es 0, ist das Granule frei.
Bei 3 Granules pro Lump sind nur die drei Bits ganz rechts relevant, die anderen 5 Bits sind immer auf 1 gesetzt.

Die folgende Tabelle veranschaulicht dies noch etwas:

Granule Nr.

7	6	5	4	3	2	1	0
1	1	1	1	1	x	x	x

Ist Bit 0 gesetzt, ist also das erste Granule dieses Lumps belegt.

Eine normale Colour-Genie Diskette benutzt aber nur 48 Lumps. In den restlichen 144 Bytes sind deshalb alle Bits gesetzt, da diese Lumps alle unbenutzbar, also "belegt" sind.

Die Bytes 192 bis 207 werden von Colour-Genie DOS nicht benutzt.

In den Bytes 208 bis 215 steht der Name der Diskette, danach folgen 8 Bytes mit dem Datum.

Die Bytes 224 bis 255 werden nicht benutzt.

Der zweite Sektor enthalt die sogenannte HIT (= Hash Index Tabelle).
In dieser Tabelle legt das Betriebssystem den in ein Byte verschlusselten Programmnamen ab.
Dies macht das Auffinden eines bestimmten Namens wesentlich einfacher und vor allen Dingen schneller, da nicht standig Texte miteinander verglichen werden mussen.
Der Aufbau der Tabelle ist auf den ersten Blick etwas kompliziert, er erklart sich jedoch durch die einfache Anwendung. Da jeder Eintrag in das Inhaltsverzeichnis 32 Bytes benotigt, passen 8 solcher Eintrage in einen Diskettensektor.
Die ersten zwolf Bytes der HIT enthalten nun die Hash-Codes der ersten Eintrage aus den Inhaltsverzeichnis-Sektoren. Die Bytes 32 bis 39 enthalten entsprechend die Hash-Codes der zweiten Eintrage eines jedes Sektors.
Ist ein Hash-Code Null, so ist der entsprechende Eintrag nicht belegt.

Eine besondere Bedeutung hat das Byte Nr. 31 (es enthalt eine 5).
Dieses Byte gibt an, wieviele Sektoren das Inhaltverzeichnis belegt. Da mindestens 10 Sektoren (zwei Granules) belegt sein mussen, wird aber nur der zusatzlich belegte Platz angegeben, also weitere 5 Sektoren.

Im dritten bis funfzehnten Sektoren stehen nun die eigentlichen Eintrage fur jedes File.
Diese bauen sich folgendermaßen auf:

1. Byte	Bit 7 = 1	: Dieser Eintrag ist ein Folgeeintrag (siehe unten)
	Bit 7 = 0	: Dieser Eintrag ist ein Ersteintrag
	Bit 6 = 1	: Dieser File ist ein Systemfile
	Bit 6 = 0	: Dieser File ist kein Systemfile
	Bit 5	: nicht benutzt
	Bit 4 = 1	: Dieser Eintrag ist belegt
	Bit 4 = 0	: Dieser Eintrag ist frei (siehe unten)
	Bit 3 = 1	: Dieser Eintrag ist unsichtbar
	Bit 3 = 0	: Dieser Eintrag ist sichtbar
	Bit 2-0	: nicht benutzt

2. Byte	Bit 7 = 1 : Dieser File darf nicht vergroßert werden. Bit 7 = 0 : Dieser File darf vergroßert werden (Normalfall) Bit 6 : nicht benutzt Bit 5 = 1 : Dieses File wurde verandert Bit 5 = 0 : Dieses File wurde nicht verandert Bit 4-0 : nicht benutzt
3. Byte	nicht benutzt
4. Byte	Die Position des ersten nicht mehr belegten Bytes im letzten benutzten Sektor des Files (Siehe unten)
5. Byte	Die Record-Lange des Files (nur wichtig bei Benutzung von Random-Access-Files)
6. bis 13. Byte	Filename, linksbundig, aufgefullt mit Leerzeichen
14. bis 16. Byte	Filetyp, linksbundig, aufgefullt mit Leerzeichen
17. bis 20. Byte	nicht benutzt
21./22. Byte	File-relative Sektornummer des letzten benutzten Sektors des Files
23. bis 30. Byte	4 Byte-Paare, die jeweils einen Bereich auf der Diskette beschreiben, der dem File zugeordnet ist: 1. Byte : - 255 (OFFH) = Fileende - 0 bis 191 (00H bis 0BFH) Nummer des Lumps, in dem der Bereich liegt

2. Byte :

- Die linken drei Bits geben an, in welchem Granule des Lumps der Bereich beginnt, die anderen funf Bits geben die Anzahl von aufeinanderfolgenden belegten Granules an (da ein Granule immer belegt ist, muß zu dieser Zahl immer 1 addiert werden)

31./32. Byte — Wenn das 31. Byte eine 254 (OFEH) enthalt, steht im 32. Byte die Nummer eines Erweiterungseintrags fur dieses File, sonst ist das File zuende.

Da jedes Bytepaar also maximal 32 Granules (= 40 KBytes) beschreiben kann, kann also jeder Eintrag maximal 128 Granules (= 160 KBytes) belegen. Ein langeres File benotigt also mehrere Eintrage. Zu diesem Zweck besteht die Moglichkeit, im 32. Byte des Eintrags einen Verweis auf einen weiteren, diesem File zugeordneten Eintrag zu geben. Dieser Verweis ist die Nummer, die dem entsprechenden Eintrag auch im HIT-Sektor zugeordnet wurde, also <Position im Sektor> * 32 + Sektornummer. Ein solcher Erweiterungseintrag sieht etwas anders aus, als ein Ersteintrag:

1. Byte — Bit 7 und Bit 4 sind 1, alle anderen Bits sind Null

2. Byte — Ruckverweis auf den vorhergegangenen Eintrag. Auf diese Weise ist eine Ruckverkettung moglich.

3. bis 22. Byte — nicht benutzt

23. bis 32. Byte — wie die entsprechenden Bytes im Ersteintrag

Besonders wichtig ist die Tatsache, daß bei Ausfuhrung des KILL-Befehls nur alle Hash-Codes aller zu diesem File gehorenden Eintrage in der HIT auf Null gesetzt werden und in jedem Eintrag das Bit 4 des ersten Bytes zuruckgesetzt wird. Ein versehentlich geloschtes Programm kann also wieder"gefunden" werden.
Hierzu ist es nur notig, in allen Eintragen das Bit 4 im ersten Byte zu setzen und an den entsprechenden Stellen in der HIT wieder den Hash-Code des Filenamens einzutragen. Dies kann zum Beispiel mit COLZAP geschehen.
Der Hash-Code eines Filenamens wird folgendermaßen berechnet:

1. Der Filename muß wie im Inhaltsverzeichnis vorliegen, also Name und Extension linksbundig und mit Leerzeichen aufgefullt.

Hier einige Beispiele dazu:

Name : READ/ME	Umgewandelt:	READ ME
Name : COLZAP/CMD	Umgewandelt:	COLZAP CMD
Name : EMPIRE	Umgewandelt:	EMPIRE

Ein umgewandelter Filename ist also immer 11 Zeichen lang. Die Berechnung des Hash-Codes für diesen Namen geschieht nun folgendermaßen:

1.Zeichen * 2 XOR 2.Zeichen * 2 XOR 3.Zeichen * 2 ...

Wenn das Ergebnis dieser Rechnung Null ist, wird 1 als Hash-Code genommen, da Null ja die Kennung eines freien Eintrags ist.

Hier einige Beispiele für Hash-Codes:

Name : READ ME	Hash-Code : 60H = 96
Name : COLZAP CMD	Hash-Code : F0H = 240
Name : EMPIRE	Hash-Code : 49H = 73

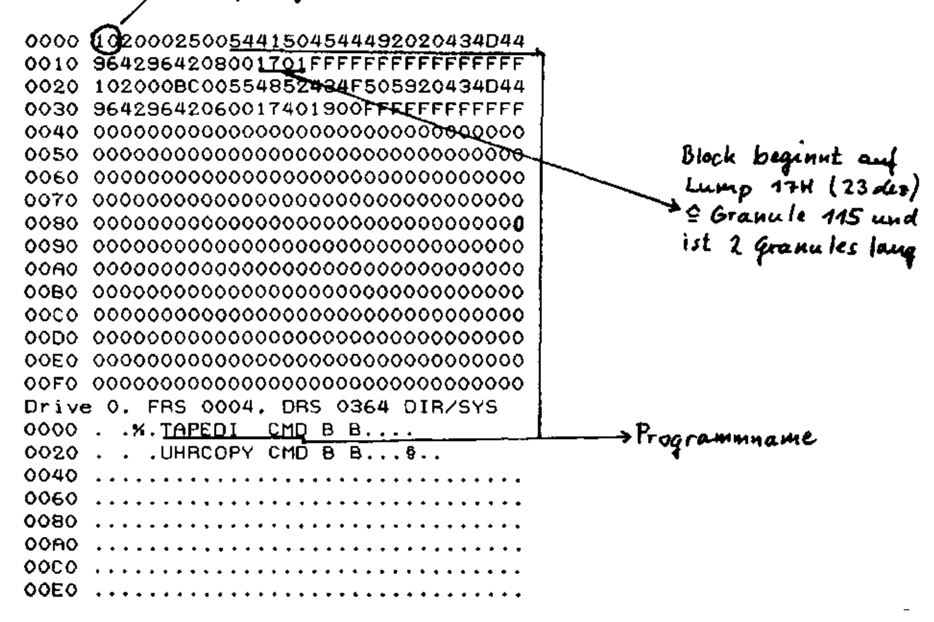
Folgendes kleine BASIC-Programm berechnet den Hash-Code eines beliebigen Filenamens:

Geben Sie zuerst den Namen ein, dann die Extension.
Der Hash-Code wird natürlich in dezimal ausgegeben; um ihn mit COLZAP u.ä. einzugeben, müssen Sie ihn noch in das Hexadezimalsystem umrechnen:

```
 10 INPUT "Filename ";A$
 20 INPUT "Extension ";B$
 30 A$=A$+STRING$(8-LEN(A$),32)+B$+STRING$(3-LEN(B$),32)
 40 A=0
 50 FOR X=1 TO 11
 60 B=ASC(MID$(A$,X,1))
 70 A=(A OR B) AND NOT(A AND B)
 80 A=2*A
 90 IF A>255 THEN A=A-255
100 NEXT X
110 IF A=0 THEN A=1
120 PRINT "Der Hash-Code ist";A
130 GOTO 10
```

```
0000 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF )
0010 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF } Disketten-Belegungs-Tabelle
0020 FFFFFFFFFFFFF9F8F8F8F8F8F8F8F8F8 )
0030 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
0040 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
0050 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
0060 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
0070 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
0080 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
0090 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
00A0 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
00B0 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
00C0 FFFFFFFFFFFFFFFFFFFFFFF820000E042
00D0 5574696C697479203232 2E30332E3834
00E0 0DFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
00F0 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Drive 0, FRS 0000, DRS 0360 DIR/SYS
0000
0020
0040
0060
0080
00A0
00C0          .. BUtility 22.03.84  --> Diskdatum
00E0 .                               --> Diskname

0000 A2C4B36FF045A900004A30AE00000000
0010 00000000000000000000000000000005
0020 0085517200050000 00DC006900000090  --> Hashcode von "TAPEDI/CMD"
0030 00000000000000000000000000000000
0040 0000006B004300000029000000000000  -->   "     "     "  "UHRCOPY/CMD"
0050 00000000000000000000000000000000
0060 000000DD00E300000000000000000000
0070 00000000000000000000000000000000  - Hash-Index-Tabelle
0080 00000038000000000000000000000000
0090 00000000000000000000000000000000
00A0 00000000000000000000000000000000
00B0 00000000000000000000000000000000
00C0 00000000000000000000000000000000
00D0 00000000000000000000000000000000
00E0 00000000000000000000000000000000
00F0 00000000000000000000000000000000
Drive 0, FRS 0001, DRS 0361 DIR/SYS
0000    o E ..J0 ...................
0020 . Qr..... .i...................
0040 ...k.C...).....................
0060 ... .  .......................
0080 ...8...........................
00A0 ...............................
00C0 ...............................
00E0 ...............................
```



```
0000 1020002500544150454449202020434D44
0010 9642964208001701FFFFFFFFFFFFFFFF
0020 102000BC00554852434F505920434D44
0030 964296420600174019OOFFFFFFFFFFFF
0040 00000000000000000000000000000000
0050 00000000000000000000000000000000
0060 00000000000000000000000000000000
0070 00000000000000000000000000000000
0080 00000000000000000000000000000000
0090 00000000000000000000000000000000
00A0 00000000000000000000000000000000
00B0 00000000000000000000000000000000
00C0 00000000000000000000000000000000
00D0 00000000000000000000000000000000
00E0 00000000000000000000000000000000
00F0 00000000000000000000000000000000
Drive 0. FRS 0004. DRS 0364 DIR/SYS
0000 . .%.TAPEDI  CMD B B....
0020 . . .UHRCOPY CMD B B...§..
0040 ................................
0060 ................................
0080 ................................
00A0 ................................
00C0 ................................
00E0 ................................
```


Auf dieser und der vorigen Seite finden Sie mit Kommentaren versehene Ausdrucke einer Disketten-Belegungs-Tabelle, einer Hash-Index-Tabelle und eines Eintragssektors des Inhaltsverzeichnisses. Diese Ausdrucke können Sie zu Hilfe nehmen, um die Erklärungen auf den vorangegangenen Seiten besser zu verstehen.

DIRSORT - alphabetisches Sortieren des Inhaltsverzeichnisses einer Diskette in BASIC

Es ist Ihnen doch bestimmt schon aufgefallen, welche Schwierigkeiten es bereitet, auf einer sehr vollen Diskette einen bestimmten Programmnamen zu finden. Es wäre doch viel einfacher, wenn alle Na-en alphabetisch sortiert wären.
Ein Programm, das diese Aufgabe für Sie übernimmt, finden Sie im folgenden beschrieben und gelistet.

Zeile 10 - 100	:	Initialisierung
Zeile 110	:	N$ enthält den Filenamen DIR/SYS:d
Zeile 120 - 200	:	Test, ob DIR/SYS auf der Diskette vorhanden ist und sich eröffnen läßt
Zeile 220	:	File als Random-Access-File eröffnen, welches nicht erweitert werden darf
Zeile 230 - 240	:	Arrays entsprechend der Inhaltsverzeichnisgröße definieren
Zeile 250	:	Einen Sektor in 8 Teile je 32 Bytes aufteilen
Zeile 260 - 320	:	Eintragssektoren einlesen und in DE$(X) ablegen
Zeile 330 - 390	:	Hash-Codes in HC(X) einlesen
Zeile 400 - 480	:	Zahl von freien, unsichtbaren und SYSTEM-Files ermitteln
Zeile 490 - 860	:	Inhaltsverzeichnisbeschreibung ausgeben
Zeile 870 - 900	:	Wenn nach NAME sortiert werden soll, sind die Bytes 6 bis 13 relevant, sonst die Bytes 14 bis 16
Zeile 910 -1120	:	Sortieren
Zeile 1130-1380	:	Inhaltsverzeichnis zurückschreiben
Zeile 1390	:	Programmende

Probelauf des Directory-Sortierprogramms:

```
Directory vorher:
>CMD"I1
NOTNAMED 00/00/00
FORTHD/CMD     RS232/CMD     BCODE2/CMD
COLCOM/CMD     COLASMD/CMD   ZEICHEN
PASCAL         TAPEDI/CMD    COLMON1/CMD
COLMON3/CMD    FORTH/CMD     ZEDIT/CMD
UHRCOPY/CMD    BASICPL5/CMD  COLOFF/CMD
COLZAP/CMD     PACKER/CMD    COLMON3D/CMD
CCOPY/CMD      EDITOR/CMD    SOUND
READY
```

```
Directory-Sortierprogramm
Welches Laufwerk soll sortiert werden? 1
Directory von Laufwerk 1
104 Directory-Eintraege :
 81 belegt,
 23 frei,
  2 unsichtbar,
  2 System-Programme
Nach <NAME> oder <EXT>ension sortieren?
NAME
```

```
Directory nachher:
>CMD"I1
NOTNAMED 00/00/00
BASICPL5/CMD BCODE2/CMD    CCOPY/CMD
COLASMD/CMD  COLCOM/CMD    COLMON1/CMD
COLMON3/CMD  COLMON3D/CMD  COLOFF/CMD
COLZAP/CMD   EDITOR/CMD    FORTH/CMD
FORTHD/CMD   PACKER/CMD    PASCAL
RS232/CMD    SOUND         TAPEDI/CMD
UHRCOPY/CMD  ZEDIT/CMD     ZEICHEN
READY
```

Programmlisting:

```
10 CLEAR 10000
20 DEFINT A-Z
30 CLS
40 COLOUR 16
50 PRINT"Directory-Sortierprogramm"
60 PRINT
70 PRINT
80 INPUT "Welches Laufwerk soll sortiert werden":A
90 A=INT(A)
100 IF A<0 OR A>3 THEN 30
110 N$="DIR/SYS:" +RIGHT$(STR$(A), 1)
120 ON ERROR GOTO 170
130 OPEN "I",1, N$
140 CLOSE
150 ON ERROR GOTO 0
160 GOTO 220
170 PRINT"Lesefehler im Directory!!!!"
180 PRINT"Druecken Sie eine Taste."
190 Z$=INKEY$
200 IF INKEY$ ="" THEN 200 ELSE 30
210 RESUME NEXT
220 OPEN "D",1, N$
230 L=( LOF(1)-2)*8
240 DIM DE$(L), HC(L)
250 FIELD 1, 32 AS H$(1), 32 AS H$(2), 32 AS H$(3), 32 AS H$(4), 32 AS
 H$(5), 32 AS H$(6), 32 AS H$(7), 32 AS H$(8)
260 FOR X=3 TO LOF(1)
270 ON ERROR GOTO 210
280 GET 1, X
290 ON ERROR GOTO 0
300 FOR Y=1 TO 8
310 DE$((LOF(1)-X)*8+Y)=H$(Y)
320 NEXT Y, X
330 ON ERROR GOTO 210
340 GET 1, 2
350 ON ERROR GOTO 0
360 FOR X=0 TO LOF(1)-3
370 FOR Y=1 TO 8
380 HC(( LOF(1)-3-X)*8+Y)=ASC(MID$(H$(Y), X+1, 1))
390 NEXT Y, X
400 FR=0
410 SS=0
420 IV=0
430 FOR X=1 TO L
440 IF ( ASC(DE$(X)) AND 16)=0 THEN FR=FR+1: GOTO 480
450 IF ( ASC(DE$(X)) AND 128)<>0 THEN 480
460 IF ( ASC(DE$(X)) AND 64)<>0 THEN SS=SS+1
```

```
470 IF ( ASC(DE$(X)) AND 8)<>0 THEN IV=IV+1
480 NEXT X
490 COLOUR 3
500 PRINT"Directory von Laufwerk ";
510 PRINT RIGHT$(N$, 1)
520 PRINT
530 COLOUR 16
540 COLOUR 2
550 PRINT USING"###";L;
560 COLOUR 5
570 PRINT" Directory-Eintraege : "
580 PRINT
590 COLOUR 2
600 PRINT USING"###";FR;
610 COLOUR 5
620 PRINT" belegt,"
630 COLOUR 6
640 PRINT STRING$(39, 217)
650 COLOUR 16
660 COLOUR 2
670 PRINT USING"###";L-FR;
680 COLOUR 5
690 PRINT" frei,"
700 COLOUR 6
710 PRINT STRING$(39, 217)
720 COLOUR 16
730 COLOUR 2
740 PRINT USING"###";IV;
750 COLOUR 5
760 PRINT" unsichtbar,"
770 COLOUR 6
780 PRINT STRING$(39, 217)
790 COLOUR 16
800 COLOUR 2
810 PRINT USING"###";SS;
820 COLOUR 5
830 PRINT" System-Programme"
840 COLOUR 16
850 PRINT
860 PRINT
870 INPUT "Nach <NAME> oder <EXT>ension sortieren";A$
880 IF A$="NAME" THEN S1=6: L1=8: S2=14: L2=3: GOTO 910
890 IF A$="EXT" THEN S1=14: L1=3: S2=6: L2=8: GOTO 910
900 GOTO 490
910 FOR X=1 TO L-1
920 IF ( ASC(DE$(X)) AND 64)<>0 THEN 1110
930 IF ( ASC(DE$(X)) AND 16)=0 THEN 1110
940 IF ( ASC(DE$(X)) AND 128)<>0 THEN 1110
950 FOR Y=X+1 TO L
960 IF ( ASC(DE$(Y)) AND 64)<>0 THEN 1100
970 IF ( ASC(DE$(Y)) AND 16)=0 THEN 1100
980 IF ( ASC(DE$(Y)) AND 128)<>0 THEN 1100
990 T1$=MID$(DE$(X), S1, L1)
1000 T2$=MID$(DE$(Y), S1, L1)
```

```
1010 IF T1$<T2$ THEN 1100
1020 IF T1$>T2$ THEN 1040
1030 IF MID$(DE$(X), S2, L2)<=MID$(DE$(Y), S2, L2) THEN 1100
1040 H$=DE$(X)
1050 H=HC(X)
1060 DE$(X)=DE$(Y)
1070 DE$(Y)=H$
1080 HC(X)=HC(Y)
1090 HC(Y)=H
1100 NEXT Y
1110 NEXT X
1120 PRINT "Sortiervorgang beendet"
1130 INPUT "Directory zurueckschreiben";A$
1140 IF A$="N" THEN 1380
1150 IF A$<>"J" THEN 1130
1160 POKE&H400C,&H00AF
1170 POKE&H400D, 201
1180 PRINT "ACHTUNG: Das Programm darf jetzt nicht"
1190 PRINT "unterbrochen werden !!!"
1200 ON ERROR GOTO 210
1210 GET 1, 2
1220 ON ERROR GOTO 0
1230 FOR X=0 TO LOF(1)-3
1240 FOR Y=1 TO 8
1250 LSET H$(Y)=LEFT$(H$(Y), X)+CHR$(HC(( LOF(1)-3-X)*8+Y))+MID$(H$(Y)
, X+2)
1260 NEXT Y, X
1270 AD=2+PEEK(&H43D0)+256*PEEK(&H43D1)
1280 POKE AD, PEEK(AD) OR 1
1290 PUT 1, 2
1300 FOR X=3 TO LOF(1)
1310 FOR Y=1 TO 8
1320 LSET H$(Y)=DE$((LOF(1)-X)*8+Y)
1330 NEXT Y
1340 POKE AD, PEEK(AD) OR 1
1350 PUT 1, X
1360 NEXT X
1370 POKE&H400C, 201
1380 CLOSE
1390 END
```

Extended Directory
Ein erweitertes Inhaltsverzeichnis in Basic

Mit diesem Programm konnen Sie mehr der Informationen sichtbar machen, die im Inhaltsverzeichnis jeder Diskette stehen. Ausgegeben werden die Lange eines jeden Programms in Granules und Bytes, die Logische Recordlange und die Position des Fileendes. Geben Sie zunachst ein, von welchem Laufwerk Sie das Inhaltsverzeichnis sehen wollen. Dann konnen Sie eine Maske fur den Filenamen eingeben, falls Sie nur einige Files angezeigt haben wollen. Diese Maske sieht prinzipiell folgendermaßen aus:

Jedes Zeichen der Maske, das kein '?' ist, muß an genau dieser Stelle im Filenamen gefunden werden, damit er unter diese Maske fallt. An Stelle eines Fragezeichens ist jedes Zeichen zugelassen. Der Name COLZAP CMD fallt also unter die Maske C?L?????C?D, der Name COLMON3DCIM jedoch nicht. Geben Sie die Maske wie einen normalen Filenamen ein, daß Auffullen mit Leerzeichen geschieht automatisch. Eine weitere Besonderheit ist noch zu erwahnen:
Geben Sie ??/CMD an, so werden nur Files angegeben, deren Name zweibuchstabig ist; geben Sie aber */CMD an, so werden alle Files, die die Extension CMD haben, angegeben.

Zum Schluß konnen Sie noch wahlen, ob alle Ausgaben auch auf den Drucker gegeben werden sollen. Mit dieser Methode wurde auch der Probeausdruck erstellt, den Sie auf der nachsten Seite sehen.

Zeile 10 - 110 : Initialisierung

Zeile 120 - 350 : Erstellung der Namens-Maske

Zeile 360 - 370 : Setzen des Druckerflags

Zeile 380 - 490 : Einlesen der Inhaltsverzeichniseintrage in DE$(X)

Zeile 500 - 1110 : Heraussuchen der passenden Namen und Ausgabe der Parameter. Nach 20 Zeilen auf Tastendruck warten

Zeile 1120 - 1490 : Berechnung und Ausgabe von Diskettenname, -datum und der Anzahl der freien Granules

Zeile 1500 - 1520 : Programmende

Probelauf des Programms 'Extended Directory:

```
Extended Directory
Welches Laufwerk? 1
Maske des Filenamens?
Alle Files
Ausgabe auch auf Drucker (J/N)? J
Name              EOF   LRL  Grans  Bytes
-----------------------------------------
BOOT/SYS          5/000 256    1     1280
DIR/SYS          15/000 256    3     3840
ZEDIT/CMD        33/136 256    7     8584
TAPEDI/CMD        8/037 256    2     2085
UHRCOPY/CMD       6/188 256    2     1724
EDITOR/CMD       36/229 256    8     9445
FORTH/CMD        46/149 256   10    11925
FORTHD/CMD       29/042 256    6     7466
PACKER/CMD       18/072 256    4     4680
RS232/CMD         1/201 256    1      457
COLZAP/CMD       22/092 256    5     5724
COLMON1/CMD      17/072 256    4     4424
COLMON3/CMD      57/088 256   12    14680
COLMON3D/CMD     59/167 256   12    15271
COLOFF/CMD        6/159 256    2     1695
COLCOM/CMD       60/166 256   13    15526
BCODE2/CMD       11/048 256    3     2864
CCOPY/CMD         2/236 256    1      748
COLASMD/CMD      26/179 256    6     6835
BASICPL5/CMD     21/102 256    5     5478
Name              EOF   LRL  Grans  Bytes
-----------------------------------------
SOUND            15/040 256    4     3880
ZEICHEN          27/220 256    6     7132


Disketten-Name : ROEDEL
Disketten-Datum: 12/34/56
  29 freie Granules
```

Programmlisting:

```
10 CLS
20 CLEAR 10000
30 DEFINT A-Z
40 COLOUR 16
50 PRINT"Extended Directory"
60 COLOUR 5
70 PRINT STRING$(18, 217)
80 COLOUR 16
90 PRINT
100 INPUT"Welches Laufwerk":A
110 N$="DIR/SYS:" +RIGHT$(STR$(A), 1)
120 A$=""
130 INPUT"Maske des Filenamens";A$
140 IF A$="" THEN M$="???????????": PRINT"Alle Files": GOTO 360
150 M1$=""
160 M2$=""
170 FOR X=1 TO LEN(A$)
180 B$=MID$(A$, X, 1)
190 IF B$="/" THEN M1$=M1$+STRING$(8-LEN(M1$), 32): X=X+1: GOTO 280
200 IF B$="*" THEN M1$=M1$+STRING$(8-LEN(M1$),"?"): GOTO 250
210 M1$=M1$+B$
220 IF X<9 THEN NEXT X: GOTO 350
230 M1$=LEFT$(M1$, 8)
240 GOTO 350
250 X=X+1
260 IF X<=LEN(A$) AND MID$(A$, X, 1)<>"/" THEN 250
270 X=X+1
280 M2$=""
290 FOR Y=X TO LEN(A$)
300 B$=MID$(A$, Y, 1)
310 IF B$="*" THEN M2$=M2$+STRING$(3-LEN(M2$),"?"): GOTO 350
320 M2$=M2$+B$
330 IF Y<X+3 THEN NEXT Y: GOTO 350
340 M2$=LEFT$(M2$, 3)
350 M$=M1$+M2$
360 INPUT"Ausgabe auch auf Drucker (J/N)";D$
370 IF D$<>"J" AND D$<>"N" THEN 360
380 OPEN "D",1, N$
390 L=( LOF(1)-2)*8
400 DIM DE$(L), H$(8)
410 FIELD 1, 32 AS H$(1), 32 AS H$(2), 32 AS H$(3), 32 AS H$(4), 32 AS
 H$(5), 32 AS H$(6), 32 AS H$(7), 32 AS H$(8)
420 FOR X=3 TO LOF(1)
430 ON ERROR GOTO 1530
440 GET 1, X
450 ON ERROR GOTO 0
460 FOR Y=1 TO 8
470 DE$((X-3)*8+Y)=H$(Y)
480 NEXT Y
490 NEXT X
500 X=1
510 CLS
```

```
520 C=0
530 X$="Name             EOF   LRL  Grans  Bytes"
540 PRINT X$
550 IF D$="J" THEN LPRINT X$
560 COLOUR 5
570 PRINT STRING$(39, 217)
580 COLOUR 16
590 IF D$="J" THEN LPRINT STRING$(39,"-")
600 T$="????????/??? xxxx/xxx xxx  xxx  xxxxxx"
610 D=ASC(DE$(X))
620 IF (D AND 128)<>0 OR (D AND 16)=0 THEN 1100
630 R$=MID$(DE$(X), 6, 11)
640 FOR Y=1 TO 11
650 IF MID$(M$, Y, 1)="?" THEN 670
660 IF MID$(M$, Y, 1)<>MID$(R$, Y, 1) THEN 1100
670 NEXT Y
680 R1$=""
690 FOR Y=1 TO 8
700 IF MID$(R$, Y, 1)<>" " THEN R1$=R1$+MID$(R$, Y, 1): NEXT Y
710 R2$=""
720 FOR Y=9 TO 11
730 IF MID$(R$, Y, 1)<>" " THEN R2$=R2$+MID$(R$, Y, 1): NEXT Y
740 R$=R1$
750 IF R2$<>"" THEN R$=R$+"/" +R2$
760 R$=R$+STRING$(12-LEN(R$), 32)
770 MID$(T$, 1, 12)=R$
780 B=ASC(MID$(DE$(X), 4, 1))
790 Z=B
800 R$=RIGHT$(STR$(B), LEN(STR$(B))-1)
810 R$=STRING$(3-LEN(R$), 48)+R$
820 MID$(T$, 19, 3)=R$
830 B=ASC(MID$(DE$(X), 5, 1))
840 IF B=0 THEN B=256
850 R$=RIGHT$(STR$(B), LEN(STR$(B))-1)
860 R$=STRING$(3-LEN(R$), 32)+R$
870 MID$(T$, 23, 3)=R$
880 B=ASC(MID$(DE$(X), 21, 1))+256*ASC(MID$(DE$(X), 22, 1))
890 R$=RIGHT$(STR$(B), LEN(STR$(B))-1)
900 R$=STRING$(4-LEN(R$), 32)+R$
910 MID$(T$, 14, 4)=R$
920 F=B+SGN(Z)
930 GR=INT(F/5)+SGN(F-INT(F/5)*5)
940 R$=RIGHT$(STR$(GR), LEN(STR$(GR))-1)
950 R$=STRING$(3-LEN(R$), 32)+R$
960 MID$(T$, 28, 3)=R$
970 KB=B*256+Z
980 R$=RIGHT$(STR$(KB), LEN(STR$(KB))-1)
990 R$=STRING$(6-LEN(R$), 32)+R$
1000 MID$(T$, 33, 6)=R$
1010 PRINT T$
1020 IF D$="J" THEN LPRINT T$
1030 C=C+1
1040 IF C<20 THEN 1100
```

```
1050 COLOUR 5
1060 INPUT"<RETURN> zum Fortfahren";X$
1070 COLOUR 16
1080 X=X+1
1090 IF X<=L THEN 510
1100 X=X+1
1110 IF X<=L THEN 610
1120 FIELD 1, 96 AS GT$, 112 AS DUMMY$, 8 AS NM$, 8 AS DT$
1130 ON ERROR GOTO 1530
1140 GET 1, 1
1150 ON ERROR GOTO 0
1160 CLOSE
1170 FR=0
1180 W=1
1190 PRINT
1200 COLOUR 2
1210 PRINT"Bitte einen Moment warten"
1220 COLOUR 16
1230 PRINT
1240 G=ASC(MID$(GT$, W, 1)) AND 7
1250 FOR Y=1 TO 3
1260 G1=INT(G/2)
1270 G2=G-G1*2
1280 FR=FR+1-G2
1290 G=G1
1300 NEXT
1310 W=W+1
1320 IF W<97 THEN 1240
1330 PRINT
1340 IF D$="J" THEN LPRINT
1350 PRINT "Disketten-Name : ";
1360 COLOUR 5
1370 PRINT NM$
1380 COLOUR 16
1390 IF D$="J" THEN LPRINT"Disketten-Name : ";NM$
1400 PRINT "Disketten-Datum: ";
1410 COLOUR 5
1420 PRINT DT$
1430 IF D$="J" THEN LPRINT"Disketten-Datum: ";DT$
1440 COLOUR 16
1450 PRINT"Freie Granules:";
1460 COLOUR 5
1470 PRINT USING"####";FR
1480 IF D$="J" THEN LPRINT USING"####";FR;: LPRINT" freie Granules"
1490 CLOSE
1500 PRINT
1510 INPUT"<RETURN> fuer Neustart";A$
1520 RUN
1530 RESUME NEXT
```

Umbenennung von Disketten

Mit diesem kleinen Basic-Programm können Sie endlich Ihren Disketten sinnvolle Namen geben, anstatt dem langweiligen NCW 1983. Starten Sie das Programm und geben Sie an, in welchem Laufwerk sich die Diskette befindet, deren Namen Sie ändern wollen. Dann werden der bisherige Name und das Datum ausgegeben. Nun können Sie einen neuen Namen und ein neues Datum angeben. Drücken Sie nur <RETURN>, so wird der alte Wert übernommen. Danach wird das Inhaltsverzeichnis zurückgeschrieben.

Programmlisting:

```
10 CLS
20 COLOUR 16
30 PRINT"Disketten Umbenennung"
40 INPUT"Welches Laufwerk ";A
50 N$="DIR/SYS:" +RIGHT$(STR$(A), 1)
60 OPEN"d",1, N$
70 FIELD 1, 208 AS DUMMY$, 8 AS NM$, 8 AS DT$
80 ON ERROR GOTO 120
90 GET 1, 1
100 ON ERROR GOTO 0
110 GOTO 130
120 RESUME NEXT
130 PRINT"Laufwerk Nr. ";RIGHT$(N$, 1)
140 PRINT
150 PRINT"Name : ";NM$; TAB(20);"Datum : ";DT$
160 A$=""
170 INPUT"Neuer Name ";A$
180 IF A$<>"" THEN LSET NM$=A$
190 A$=""
200 INPUT"Neues Datum ";A$
210 IF A$<>"" THEN LSET DT$=A$
220 AD=2+PEEK(&H43D0)+256*PEEK(&H43D1)
230 POKE AD, PEEK(AD)OR 1
240 PUT 1, 1
250 CLOSE
260 END
```

Anhang A: Basic-Token Liste

Basic Einfachtokens

Code	Token
128	END
129	FOR
130	RESET
131	SET
132	CLS
133	CMD
134	RANDOM
135	NEXT
136	DATA
137	INPUT
138	DIM
139	READ
140	LET
141	GOTO
142	RUN
143	IF
144	RESTORE
145	GOSUB
146	RETURN
147	REM
148	STOP
149	ELSE
150	TRON
151	TROFF
152	DEFSTR
153	DEFINT
154	DEFSNG
155	DEFDBL
156	LINE
157	EDIT
158	ERROR
159	RESUME
160	OUT
161	ON
162	OPEN
163	FIELD
164	GET
165	PUT
166	CLOSE
167	LOAD
168	MERGE
169	NAME
170	KILL
171	LSET
172	RSET
173	SAVE
174	SYSTEM
175	LPRINT
176	DEF
177	POKE
178	PRINT
179	CONT
180	LIST
181	LLIST
182	DELETE
183	AUTO
184	CLEAR
185	CLOAD
186	CSAVE
187	NEW
188	TAB(
189	TO
190	FN
191	USING
192	VARPTR
193	USR
194	ERL
195	ERR
196	STRING$
197	INSTR
198	CHECK
199	TIME$
200	MEM
201	INKEY$
202	THEN
203	NOT
204	STEP
205	+
206	-
207	*
208	/
209	↑
210	AND
211	OR
212	>
213	=
214	<
215	SGN
216	INT
217	ABS
218	FRE
219	INP
220	POS
221	SQR
222	RND
223	LOG
224	EXP
225	COS
226	SIN
227	TAN
228	ATN
229	PEEK
230	CVI
231	CVS
232	CVD
233	EOF
234	LOC
235	LOF
236	MKI$
237	MKS$
238	MKD$
239	CINT
240	CSNG
241	CDBL
242	FIX
243	LEN
244	STR$
245	VAL
246	ASC
247	CHR$
248	LEFT$
249	RIGHT$
250	MID$
251	'

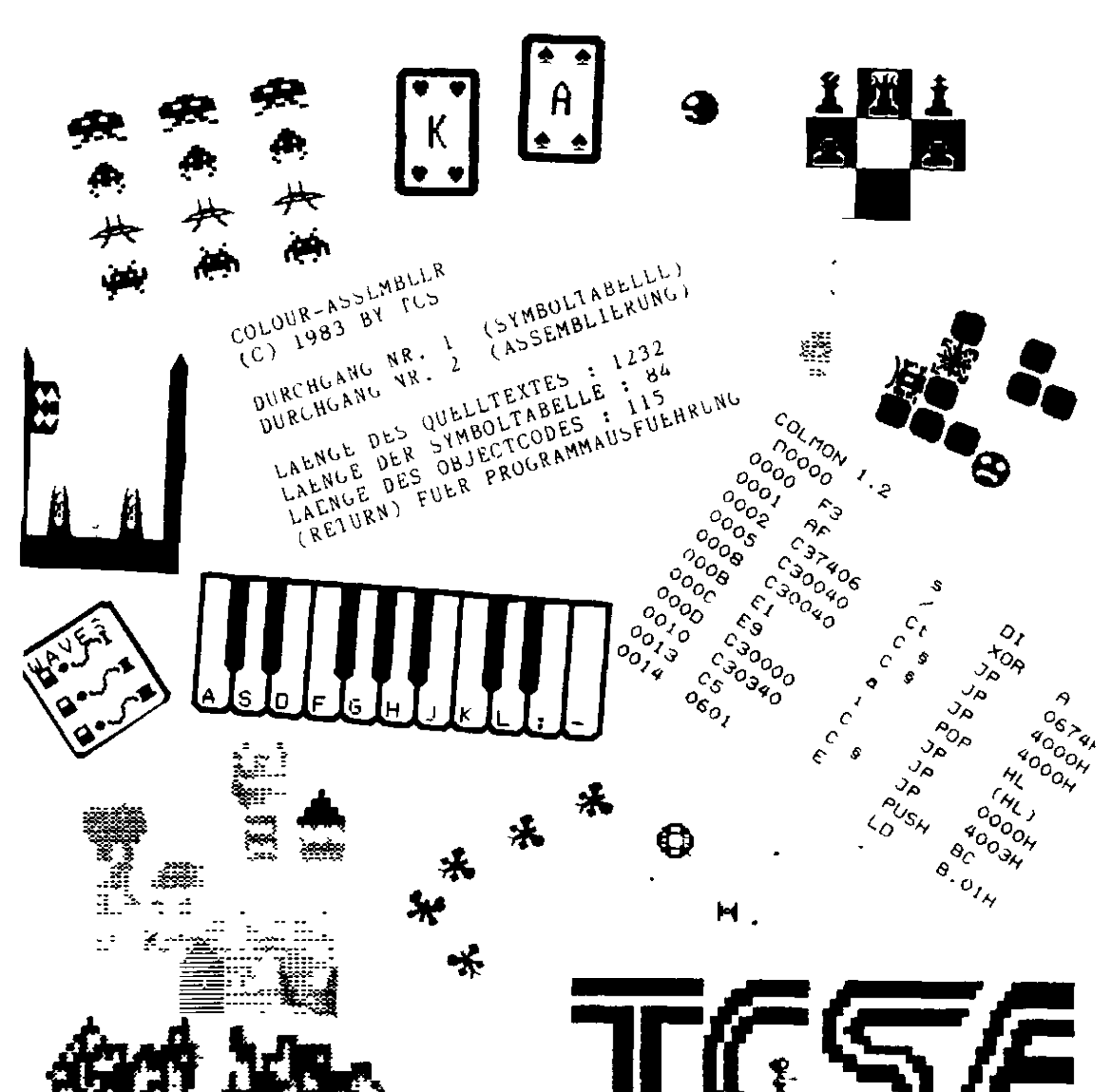
Colour-Basic Doppeltokens

255	128		COLOUR	255	129		FCOLOUR
255	130		KEYPAD	255	131		JOY
255	132		PLOT	255	133		FGR
255	134		LGR	255	135		FCLS
255	136		PLAY	255	137		CIRCLE
255	138		SCALE	255	139		SHAPE
255	140		NSHAPE	255	141		XSHAPE
255	142		PAINT	255	143		CPOINT
255	144		NPLOT	255	145		SOUND
255	146		CHAR	255	147		RENUM
255	148		SWAP	255	149		FKEY
255	150		CALL	255	151		VERIFY
255	152		BGRD	255	153		NBGRD

0 =	00H		1 =	01H		2 =	02H	
3 =	03H		4 =	04H		5 =	05H	
6 =	06H		7 =	07H		8 =	08H	
9 =	09H		10 =	0AH		11 =	0BH	
12 =	0CH		13 =	0DH		14 =	0EH	
15 =	0FH		16 =	10H		17 =	11H	
18 =	12H		19 =	13H		20 =	14H	
21 =	15H		22 =	16H		23 =	17H	
24 =	18H		25 =	19H		26 =	1AH	
27 =	1BH		28 =	1CH		29 =	1DH	
30 =	1EH		31 =	1FH		32 =	20H	
33 =	21H	!	34 =	22H	"	35 =	23H	#
36 =	24H	$	37 =	25H	%	38 =	26H	&
39 =	27H	'	40 =	28H	(	41 =	29H	)
42 =	2AH	*	43 =	2BH	+	44 =	2CH	,
45 =	2DH	-	46 =	2EH	.	47 =	2FH	/
48 =	30H	0	49 =	31H	1	50 =	32H	2
51 =	33H	3	52 =	34H	4	53 =	35H	5
54 =	36H	6	55 =	37H	7	56 =	38H	8
57 =	39H	9	58 =	3AH	:	59 =	3BH	;
60 =	3CH	<	61 =	3DH	=	62 =	3EH	>
63 =	3FH	?	64 =	40H	@	65 =	41H	A
66 =	42H	B	67 =	43H	C	68 =	44H	D
69 =	45H	E	70 =	46H	F	71 =	47H	G
72 =	48H	H	73 =	49H	I	74 =	4AH	J
75 =	4BH	K	76 =	4CH	L	77 =	4DH	M
78 =	4EH	N	79 =	4FH	O	80 =	50H	P
81 =	51H	Q	82 =	52H	R	83 =	53H	S
84 =	54H	T	85 =	55H	U	86 =	56H	V
87 =	57H	W	88 =	58H	X	89 =	59H	Y
90 =	5AH	Z	91 =	5BH	[	92 =	5CH	\
93 =	5DH	]	94 =	5EH	^	95 =	5FH	_
96 =	60H	`	97 =	61H	a	98 =	62H	b
99 =	63H	c	100 =	64H	d	101 =	65H	e
102 =	66H	f	103 =	67H	g	104 =	68H	h
105 =	69H	i	106 =	6AH	j	107 =	6BH	k
108 =	6CH	l	109 =	6DH	m	110 =	6EH	n
111 =	6FH	o	112 =	70H	p	113 =	71H	q
114 =	72H	r	115 =	73H	s	116 =	74H	t
117 =	75H	u	118 =	76H	v	119 =	77H	w
120 =	78H	x	121 =	79H	y	122 =	7AH	z
123 =	7BH	{	124 =	7CH	|	125 =	7DH	}
126 =	7EH	~	127 =	7FH	█	128 =	80H	
129 =	81H		130 =	82H		131 =	83H	
132 =	84H		133 =	85H		134 =	86H	
135 =	87H		136 =	88H		137 =	89H	
138 =	8AH		139 =	8BH		140 =	8CH	
141 =	8DH		142 =	8EH		143 =	8FH	
144 =	90H		145 =	91H		146 =	92H	
147 =	93H		148 =	94H		149 =	95H	
150 =	96H		151 =	97H		152 =	98H	
153 =	99H		154 =	9AH		155 =	9BH	
156 =	9CH		157 =	9DH		158 =	9EH	
159 =	9FH		160 =	A0H		161 =	A1H	
162 =	A2H		163 =	A3H		164 =	A4H	
165 =	A5H		166 =	A6H		167 =	A7H	
168 =	A8H		169 =	A9H		170 =	AAH	
171 =	ABH		172 =	ACH		173 =	ADH	
174 =	AEH		175 =	AFH		176 =	B0H	
177 =	B1H		178 =	B2H		179 =	B3H	
180 =	B4H		181 =	B5H		182 =	B6H	
183 =	B7H		184 =	B8H		185 =	B9H	
186 =	BAH		187 =	BBH		188 =	BCH	
189 =	BDH		190 =	BEH		191 =	BFH	
192 =	C0H	◆	193 =	C1H	≡	194 =	C2H	┼
195 =	C3H	▒	196 =	C4H	▒	197 =	C5H	╳
198 =	C6H	♥	199 =	C7H	⧓	200 =	C8H	♠
201 =	C9H	╬	202 =	CAH	█	203 =	CBH	◘
204 =	CCH	╥	205 =	CDH	◢	206 =	CEH	▌

207 = CFH	208 = D0H	209 = D1H
210 = D2H	211 = D3H	212 = D4H
213 = D5H	214 = D6H	215 = D7H
216 = D8H	217 = D9H	218 = DAH
219 = DBH	220 = DCH	221 = DDH
222 = DEH	223 = DFH	224 = E0H
225 = E1H	226 = E2H	227 = E3H
228 = E4H	229 = E5H	230 = E6H
231 = E7H	232 = E8H	233 = E9H
234 = EAH	235 = EBH	236 = ECH
237 = EDH	238 = EEH	239 = EFH
240 = F0H	241 = F1H	242 = F2H
243 = F3H	244 = F4H	245 = F5H
246 = F6H	247 = F7H	248 = F8H
249 = F9H	250 = FAH	251 = FBH
252 = FCH	253 = FDH	254 = FEH
255 = FFH		

Colour-Genie
Software



****

Verzeichnis der verfügbaren Colour-Genie-Software

****

1.) HILFSPROGRAMME, PROGRAMMIERSPRACHEN

COLOUR-COMPILER: Dieses Programm ist extrem wertvoll. Es bietet die Möglichkeit, Basicprogramme in Maschinenspracheprogramme zu übersetzen.
Der wesentliche Vorteil: Ihr Programm läuft nach Compilierung c.a. 40 mal (!) schneller.
Dies geschieht interaktiv, d.h. Basicprogramm, Maschinenprogramm und Compiler stehen gleichzeitig im Speicher, was ein sehr schnelles Arbeiten ermöglicht. Dabei können c.a. 9.5 kByte Basicprogramm verarbeitet werden. Verzichtet man auf die hochauflösende Grafik, steht noch mehr Speicher zur Verfügung.
Voraussetzung: Ihr COLOUR-GENIE MUSS AUF 32K RAM ERWEITERT SEIN. Die meisten Colour-Basic-Befehle werden vom Colour-Compiler unterstützt. Die wesentlichste Einschränkung ist, daß der Colour-Compiler nur Ganzzahlen (Integers) verarbeitet.
Preis: 69.- DM   Diskversion: 95.- DM

COLOUR-MONITOR I: Ein Maschinensprachemonitor mit starkem Befehlssatz, z.B. Disassemlieren, Ascii/Hex-Dump, Bänder laden/schreiben, Bytes suchen, Speicher editieren/verschieben/relozieren, Hex/Dezimal-Umwandlung u.a.m.
Preis: 39.- DM

COLOUR-ASSEMBLER: Mit diesem Programm können Maschinenspracheprogramme auf komfortable Weise entwickelt werden. Das Programmieren geschieht mit Z80-Mnemonics, Labels etc.
Der Assembler erzeugt dann das Maschinenspracheprogramm, das zur Probe auch direkt ausgeführt werden kann.
Eine weitere Besonderheit des Assemblers ist, daß der Assembler-Text im ganz normalen Basic-Modus erzeugt wird, so daß alle Basic-Befehle wie EDIT, LLIST, CSAVE vom Assembler ausgenutzt werden.
Preis: 69.- DM

SOUND-EDITOR (auch im Handbuch "Colour Basic leicht gelernt" aufgelistet) : Ein nützliches Hilfsprogramm zur Programmierung des PSG-ICs (Sound-Chip). Sie editieren die PSG-Register auf dem Bildschirm und der entsprechende Ton wird gleichzeitig ausgegeben. Abschließend gibt das Programm alle PSG-Registerinhalte in dezimaler Schreibweise aus.
Preis: 25.- DM

ZEICHEN-EDITOR (auch im Handbuch "Colour Basic leicht gelernt" aufgelistet) : Dieses Programm ermöglicht es, 64 der 128 frei definierbaren Zeichen auf dem Bildschirm übersichtlich zu editieren. Es können auch Grafiken, wie z.B eine Schreibschrift, erstellt werden. Abschließend werden die definierten Zeichen in ein Basicprogramm geschrieben, was diese für den späteren Gebrauch oder zur Entwicklung eigener Programme wieder definiert. So wird die umständliche Handhabung der Programmierung der definierbaren Zeichen umgangen.
Preis: 25.- DM

GRAFIK-EDITOR: Wenn Sie Ihr Colour-Genie auf 32K RAM aufgerüstet haben, können Sie dieses Programm einsetzten, denn der "Grafik-Editor" hat eine Länge von 26000 Bytes!
Er bietet die Möglichkeit, Grafiken im FGR-Modus mit einer Vielzahl von leistungsstarken Kommandos zu kreieren. Abschließend wird ein Basic-Programm erzeugt werden, das, eingebunden in ein eigenes Programm, das Bild blitzschnell wieder auf den Bildschirm bringt. Sie können also auch komplexe Grafiken sehr schnell fertigstellen und sie hinterher beliebig verwenden.
Preis: 69.- DM

SHAPER: Die Handhabung des Basicbefehls "SHAPE" ist recht umständlich. Hier bringt der "Shaper" Abhilfe: Mit wenigen Tastendrücken erzeugen Sie eine Figur, die abschließend im Foramt der "Shape-Table" abgespeichert wird.
Preis: 25.- DM

COLROT: Ein sehr nützliches Grafikprogramm. für alle die mit großen Texten Aufmerksamkeit erregen wollen (z.B. im Schaufenster). Colrot erzeugt Laufschriften mit bis zu 512 Zeichen Länge, Titel und Fußschrift, Inversdarstellung, Intermezzo u.v.a.m. Neu:Texte können auf Band gespeichert werden. Ein komfortabler Editor ermöglicht einfaches Arbeiten.
Preis: 69.- DM

BASICODE 2 (!!): Der neue Basicode-Standard für das Colour-Genie (kompatibel zu Basicode 1). Dieses Programm macht ihr Colour-Genie Basicode kompatibel !
Basicode ist das Standard-Basic, in dem z.B. das WDR-Fernsehen Programme austrahlt. Eine Hardwareänderung ist nicht nötig.
Preis: 25.- DM

ZEICHENEDITOR+: Dieser Editor, in Maschinensprache geschrieben, ermöglicht es alle 128 Zeichen sehr schnell und komfortabel zu editieren. Dabei gibt es sogar Befehle um Kreise zu zeichnen, Zeichen zu kopieren/duplizieren/rotieren/invertieren u.v.a.m.
Preis: 39.- DM

BASIC+5: Dieses Programm erweitert das normale Basic um 5 Befehle: Im Grafikmodus können beliebige Texte dargestellt werden, Rechteckflächen können gemalt werden, mit "SAVE" können Machinenspracheprogramme gesichert und mit "LOAD" komfortabel geladen werden. Ferner steht ein Eingabebefehl zur Verfügung, der an beliebige Speicherzellen schreibt.
Preis: 39.- DM

BASIC-PACKER: Der "Packer" ist ein äußerst nützliches Hilfprogramm für alle, die in Basic programmieren. Die effektivste Funktion dieses Programms ist das "Packen", d.h. ein Basicprogramm wird so stark wie möglich gekürzt, ohne daß die Programmlogik verändert wird. Dies wird erreicht durch das Entfernen von Leerzeichen und (auf Wunsch) REMs, sowie dadurch, daß so viele Befehle wie möglich in eine Zeile gebracht werden. Vorteil: Das Programm wird schneller und kürzer. Ferner kann der "Packer" entpacken, kürzen ohne Zeilennummeränderungen, renumbern und Zeilen verschieben. (32K RAM erforderlich!)
Preis: 39.- DM

ZEICHENSATZE: Dieses Programm laßt Sie den vorhandenen Zeichensatz fur alle ASCII-Zeichen wahlweise gegen einen von acht neuen, interessanten Zeichensatzen ersetzen.
Fur dieses Programm muß Ihr Colour-Genie mit 32K RAM ausgerustet sein.
Preis: 25.- DM

SCREEN-PRINTER: (Auch in dem Buch "Das Colour-Genie-Buch 1" aufgelistet.) Auf dieses Programm haben die Besitzer des STAR-Druckers DP 510 bzw. DP 515 sicher gewartet !
Es ermoglicht es Ihnen, den Bildschirm jederzeit komplett auf Ihren Drucker auszugeben, egal ob Sie im FGR- oder im LGR-Modus sind. Definierte und feste Grafikzeichen werden ebenfalls mit ausgedruckt. Auch fur dieses Programm brauchen Sie 32K RAM und naturlich einen STAR DP 510 / DP 515 Drucker.
Preis: 39.- DM

COLOUR-MONITOR III: Dieses Programm wird alle, die auf Maschinensprache-Ebene programmieren begeistern !
Der "Colour-Monitor III" ist nicht nur ein Monitor mit den ublichen Befehlen, wie ASCII-Dump, Hex-Dump, Disassemblieren, Bander schreiben etc., sondern ein Monitor mit eingebautem Assembler, Find-Assembler und der Moglichkeit Maschinenspracheprogramme in Einzelschrittausfuhrung zu testen ! Das Arbeiten wird neben dem Super-Befehlssatz durch raffinierte Aufteilung des Bildschirms erleichtert.
Fur dieses Programm benotigen Sie ein Colour-Genie mit 32K Ram.
Ein ausfuhrliches deutsches Handbuch liegt bei.
Preis: 95.- DM DISKVERSION: 105.- DM

COLOUR-FORTH: Endlich gibt es eine leistungsstarke Version der Forth-Programmiersprache auch fur das Colour-Genie!
"Colour-Forth" ist eine fur Kassettenbetrieb angepaßte Version des FIG-Forth, die um viele Befehle fur das Colour-Genie erweitert wurde (Graphik, Ton...). Der wesentliche Vorteil der Programmiersprache Forth ist die extrem schnelle Ausfuhrungszeit der Programme, die in unserem Forth noch durch Benutzung des eingebauten Assemblers gesteigert werden kann !
Fur dieses Programm benotigen Sie 32K Ram, sowie die neuen Basic-Roms, die seit April '83 in alle Colour-Genies eingebaut sind. Ein ausfuhrliches deutsches Handbuch wird mitgeliefert.
Preis: 95.- DM DISKVERSION: 105.- DM

COLOUR-TEXT: Endlich konnen alle Colour-Genie-Besitzer, die einen Drucker haben in den Genuß einer komfortablen Textverarbeitung kommen ! Colour-Text erweitert die Bildschirmbreite auf 55 Zeichen, die maximale Zeilenlange ist 64 Zeichen (horizontales Scrolling). Sie sind mit dem Cursor standig im Text, und keine Steuerzeichen verunstalten den Text, da die verschiedenen Schriftarten durch Farben gekennzeichnet sind. (Fur Benutzer eines Schwarz/Weiss-Sichtgerates extra Statuszeile.) Das Programm unterstutzt alle gangigen Drucker (STAR, EPSON, ITOH, Typenrad). Wenn Sie einen grafikfahigen STAR-, EPSON- oder ITOH-Drucker haben konnen Sie sogar selbstdefinierte Grafikzeichen in den Text einfugen !
Ein deutsches Handbuch liegt bei.
Preis: 95.- DM DISKVERSION: 105.- DM

2.) SPIELE und GRAFIKPROGRAMME

INVASION AUS DEM WELTRAUM: Ein Maschinenspracheprogramm, das die Sound- u. Grafikmoglichkeiten des Colour-Genies voll ausnutzt. Ihre Aufgabe ist es, einen Pulk von Invasoren abzuwehren, der sich der Erde nahert.
Die Invasoren werden dabei immer schneller und gefahrlicher.
Preis: 39.- DM

PUNKTEJAGD: Bei diesem Spiel geht es darum, alle Punkte in einem Labyrint aufzusammeln, bevor Sie von einem Wachter eingeholt werden. Schnelle Grafik und Ton durch Maschinensprache.
Preis: 25.- DM

WURM: Unser derzeit schnellstes Action-Spiel. Ein Wurm kommt von oben den Bildschirm herab und versucht Sie zu vernichten. Dabei hat er die Spinne, die Fliege und die Ente als Helfer. Kein Spiel fur ruhige Stunden!
Preis: 39.- DM

BREAK OUT: Bei diesem Video-Spiel muß eine Mauer mit Ihrem Ball zerstort werden. Dabei konnen verschieden Schwierigkeitsgrade vorgewahlt werden. Hohe Geschwindigkeit durch Maschinenspracheprogrammierung.
Preis: 39.- DM

DEMOPROGRAMM: Hiermit konnen Sie besser als mit der mit Ihrem Colour-Genie z.Zt. mitgelieferten englischen Demokassette die Moglichkeiten Ihres Colour-Genies demonstrieren.
Preis: 25.- DM

ANDROMEDA: Ein erstklassiges dreidimensionales Weltraumspiel mit ansprechender Grafik. Feindliche Raumschiffe kommen auf Sie zu; vernichten Sie diese, bevor sie Ihnen wertvolle Energie abgesaugt haben. Ist die Zeit abgelaufen, so eilt Ihnen Ihre Mutterbasis zu Hilfe; aber das Auftanken will auch gelernt sein.
Preis: 39.- DM

MAU-MAU: Endlich haben Sie einen Spielpartner, der nicht wutend die Karten wegwirft, wenn er am verlieren ist. Dies durfte wohl aber auch selten der Fall sein, vielmehr wahrscheinlich ist es, daß Sie den Computer vor Wut ausschalten, da ihr Computer Dank einer hervorragenden Taktik auf Sieg programmiert ist.
(Er schummelt nicht !!) Gespielt wird nach den ublichen Regeln.
Preis: 25.- DM

HEKTIK: Stellen Sie sich vor, Sie waren in einem Neubau mit 6 Geschossen, bei dem die Ebenen durch Leitern verbunden sind. Jetzt kommen Ihnen von oben Verfolger entgegen. Sie mußen sich ihnen stellen, denn es gibt keinen Fluchtweg. Graben Sie an strategisch wichtigen Stellen Locher in den Boden, um so Ihre Verfolger auszuschalten.
Preis: 39.- DM

METEOR: Ein Super-Action-Spiel im Grafik-Modus Ihres ColourGenies. Sie befinden sich in einem Meteoritenfeld und werden von Meteoriten, Sternen und schiessenden Ufos bedrangt. Das Programm zeichnet sich durch sehr schnelle Grafik und gute Toneffekte aus. Preis: 69.- DM

MOTTEN: Bei diesem Videospiel mußen Sie Kolonnen von Motten bekampfen, die in gefahrlichen Sturzflugen anfliegen und dabei auch noch schiessen.
Preis: 39.- DM

PANIK: Ein Programm fur 2 Spieler. Man muß versuchen, den Gegner einzumauern. Wahlbare Geschwindigkeit macht "Mauer" entweder zu einem Reaktions- oder zu einem Strategiespiel. Das Programm kann wahlweise mit Joysticks oder mit der Tastatur bedient werden.
Preis: 25.- DM

TAUSENDFUß: Hier sollen Sie einen Wurm so steuern, daß er nur Futter aber kein Gift frißt. Dazu kommen noch Kraftfutter und Gegengift. Ein sehr unterhaltsames Geschicklichkeitsspiel fur die ganze Familie. Bei Spielbeginn konnen viele verschiedene Geschwindigkeiten und Schwierigkeitsstufen vorgewahlt werden.
Preis: 39.- DM

COLOUR-SCHACH: Jetzt konnen Sie gegen Ihr Colour-Genie auch Schach spielen. Colour-Schach bietet 4 verschiedene Spielstarken, Aufzeichnung eines laufenden Spieles auf Kassette, Andern von Stellungen, einen Demonstrationsmodus und anderes mehr. Die Figuren werden auf ansprechende Weise grafisch dargestellt.
Preis: 69.- DM

EXNIMROID: Ein Denkspiel, abgeleitet vom bekannten Nim-Spiel. Es geht darum, aus mehreren Haufchen den letzten Spielstein zu nehmen. Sie spielen gegen den Computer.
Preis: 25.- DM

KINGS: Eine Regierungssimulation, bei der Sie fur eine Dauer von 8 Jahren eine Insel regieren sollen. Dabei kann nur der kluge Einsatz aller Faktoren zu einem Erfolg fuhren. Haben Sie Ihre Amtszeit uberlebt, werden Ihre Taten ausgewertet, und Sie konnen dann auch weiter regieren.
Preis: 25.- DM

CONQUER: Ein spannendes Videospiel - erobern Sie die Spinnenfestung !
Preis: 25.- DM

JOKER-POKER: (Autor: J. Buchmuller) Pokern Sie gegen Ihr Colour-Genie ! Tolle Grafik und guter Sound.
Preis: 39.- DM

MAMPFMAN: Ahnlich wie unser Spiel "Punktejagd", nur daß Sie hier von mehreren Gespenstern verfolgt werden, die Sie unter bestimmten Bedingungen jedoch auch fressen konnen.
Preis: 25.- DM

MAMPFMAN 2: Die neue Version von Mampfman ! Excellente Grafik und hohe Spieldynamik. Ein sehr empfehlenswertes Programm !
Preis: 39.- DM

COLOUR-KONG: Das absolute Videospiel ! Retten Sie Ihre Freundin vor dem wilden Affen. Mehrere Ebenen, super Grafik und Musik. Auf dem Band finden Sie Versionen fur 32K und 16K RAM
Preis: 69.- DM

PANZERKAMPF: Ein Spiel fur zwei Personen (JOYSTICKS und 32K RAM erforderlich). Liefern Sie sich mit Ihrem Gegner spannende Panzerschlachten ! Das Programm verfugt uber 3 verschiedene Gelande mit Minen etc.
Preis: 69.- DM

EIS: Ein spannendes, zugleich jedoch auch recht schwieriges Actionspiel mit strategischen Elementen. Sie befinden sich in einem Kuhlraum und mußen Eisblocke so ans Rutschen bringen, daß die gefahrlichen Schneemonster vernichtet werden.
Preis: 39.- DM

EAGLE: Vernichten Sie verschiedene Schwadronen von Vogeln, bevor Sie versuchen das große Mutterschiff zu zerstoren.
Preis: 39.- DM

GAME of LIFE: Dieses bekannte Programm simuliert das Wachstum von Bakterienkulturen. Die Regeln werden im Programm erklart und einige interessante Figuren sind fest abgespeichert.
Preis: 25.- DM

SAUG: Ein lustiges Videospiel, bei dem Sie aus einem unterirdischen Labyrint Termiten ansaugen mußen, wobei Sie sich vor giftigen, roten Termiten huten mußen.
Das Spiel kann auch mit Joysticks gesteuert werden.
Preis: 39.- DM

ExREVERSIC: Spielen Sie Reversi gegen Ihr Colour-Genie !
Dabei kann zwischen mehreren Spielstufen gewahlt werden.
Das Spielbrett wird auf ansprechende Weise graphisch dargestellt.
Preis: 39.- DM

BANG-BANG: Zwei Cowboys begegnen sich in der Prarie und liefern sich ein erbittertes Duell. Fur dieses Spiel sind Joysticks erforderlich.
Preis: 39.- DM

NETZO: Versuchen Sie mit Ihrem Pinsel alle Flachen auszumalen, ohne von Ihren Gegnern erwischt zu werden.
Preis: 39.- DM

DEATH-TRAP, ein dreidimensionales Grafik-Abenteuerspiel (Adventure):
Ein Programm zum wahnsinnig werden! Sie befinden sich in einem Labyrinth mit uber 1100 Raumen, in dem es von gefahrlichen Einwohnern und Gegenstanden wimmelt. Finden Sie die wichtigen Gegenstande und wenden Sie sie richtig an, um aus dem Labyrinth zu entkommen. Zwei Vorraussetzungen mussen allerdings erfullt sein: Ihr Colour-Genie muß 32K RAM haben, und Sie mussen elementare Englischkenntnisse besitzen, da die Sie Kommandos in Form von englischen Satzen eingeben.
Preis: 69.- DM

TCS-CHOPPER: Die Geschmacker sind verschieden - aber dies ist wohl das beste Videospiel, das es z.Zt. fur das Colour-Genie gibt! Dreizehn Ihrer Kameraden sind in einem von vier riesigen Labyrinthen ausgesetzt worden. Versuchen Sie nun, diese mit Ihrem Hubschrauber zu retten! Aber Vorsicht, Ihre Freunde werden scharf bewacht!
32K RAM sind auch hier notwendig.
Preis: 69.- DM

DOPPEL-WURM: Vielleicht kennen Sie unser beliebtes Spiel "Tausendfuß". "Doppel-Wurm" ist sehr ahnlich, allerdings spielen hier zwei Spieler gegeneinander, was die Spannung naturlich erhoht. Achtung: Nur fur Joystick-Besitzer !
Preis: 39.- DM

MADTREE: Ein schwieriges Videospiel: Sie sind eine fleißige Biene und mussen eine Reihe von Blumen standig bestauben, damit diese nicht eingehen. Doch diese Blumen sind undankbar und gefahrden Sie.
Preis: 39.- DM

DIG-BOY: In Ihrem unterirdischen Reich kampfen Sie gegen Drachen und andere Untiere. Locken Sie diese unter einen der wackligen Felsen, um sie zu zerschmettern.
Dieses Spiel ist sehr empfehlenswert, da es eine Menge Abwechslung bietet.
Preis: 39.- DM

EMPIRE: Fur alle, die nicht nur reine Aktion wollen, ist "Empire" das ideale Spiel. Dieses Spiel kann mit bis zu sechs Spielern gleichzeitig gespielt werden, der Computer spielt ebenfalls mit. Jeder Spieler hat ein kleines Reich, das nun durch Handel, Politik, Krieg u.s.w. zu Wohlstand kommen soll. Dabei stehen Sie mit den anderen Nationen standig in Beziehung, sei es durch Handel oder durch Krieg.
Ansprechende Grafik macht das Programm noch interessanter.
Ihr Colour-Genie muß fur "Empire" auf 32K RAM aufgerustet sein.
Preis: 69.- DM

LUNAR-LANDER: Eine perfekte Mondlandungssimulation mit Supergrafik ! Die Idee ist zwar alt, aber die Ausfuhrung dieses Programmes macht den "Lunar-Lander" zu einem unserer besten Spiele. Sie benotigen 32K Ram. Das Spiel kann wahlweise auch mit den EG2013-Joysticks gesteuert werden.
Preis: 39.- DM

COLOUR-FROGGER: Retten Sie Ihren Frosch zuerst uber die Straße und dann uber den Fluß ! Ein spannendes Unterhaltungsspiel, an dem die ganze Familie lange Zeit Spass haben wird. Besonders gelungen ist bei diesem Programm die musiklische Untermalung.
Preis: 39.- DM

CRAZY-PAINT: Das neueste Spiel von J. Buchmuller, dem Author des TCS-Choppers - Super-Grafik und Super-Ton !
In 16 Levels mussen Sie, ahnlich wie bei unserem Spiel 'Netzo', versuchen alle Flachen auszumalen, ohne sich von den hektischen Gegnern erwischen zu lassen.
Preis: 69.- DM

MEMORY: Das beliebte Brettspiel nun auch als Version für das Colour-Genie !
Schone Grafik und die gute Spielidee machen dieses Programm empfehlenswert.
Preis: 39.- DM

GALACTIC ATTACK: Das neue Spiel von Klaus Weber. Versuchen Sie eine Vielzahl wilder Angreifer so lange wie moglich zuruckzuhalten. Ein sehr schnelles Spiel mit guter Grafik.
Preis: 39.- DM

COLOUR-SKAT: Jetzt konnen Sie auch Skat gegen Ihr Colour-Genie spielen ! Wenn Sie mal Lust zum Skat spielen haben, Ihnen aber die Spielpartner fehlen, spielen Sie halt gegen den Computer !
Gute Grafik. Achtung: 32K RAM sind notwendig.
Preis: 39.- DM

Colour-Genie - FLUGSIMULATOR: Endlich ist er da !!!!
Der Colour-Genie Flugsimulator bietet dreidimensionale Dartellung in Echtzeit, Radar, 18 Instrumente, Luftkampfmodus ...
Ein Programm, daß jeder Colour-Genie-Besitzer haben sollte !
Eine ausfuhrliche deutsche Anleitung liegt bei.
Achtung: Das Programm benotigt 32K RAM.
Preis: 69.- DM

3.) MUSIKPROGRAMME

MUSIK: Dieses Programm spielt festprogrammierte Musikstücke mit verblüffender Qualität.
Preis: 25.- DM

ORGEL: Spielen Sie Orgel auf Ihrem Colour-Genie !
Ihnen stehen zwei Manuale, einstellbares Delay und sogar eine Schlagzeugbegleitung zur Seite.
Preis: 25.- DM

COLOUR-SYNTHESIZER: Ein Programm, das man gesehen haben muß, um es überhaupt für möglich zu halten. Der Colour-Synthesizer macht aus Ihrem Colour-Genie einen vollwertigen 3-Kanal Synthesizer mit VCO, VCA, Hüllkurve, Schlagzeug...
Acht Einstellungen aller Regler können vorprogrammiert, auf Tastendruck abgerufen und auf Band gesichert werden.
Verblüffend ist auch die graphische Darstellung.
Preis: 69.- DM

COMPOSER: Mit diesem Programm können Sie sehr einfach eigene Musikstücke komponieren. Noten, Pausenzeichen u.s.w. werden grafisch dargestellt, können editiert und auf Band gespeichert werden.
Preis: 39.- DM

4.) LERNPROGRAMME ELEKTRONIKPROGRAMME

MATHEMATIK-LERNPROGRAMM: Interessant fur Kinder im Grundschulalter. Wahlweise konnen die vier Grundrechenarten geubt werden, wobei man zwischen verschiedenen Schwierigkeitsgraden wahlen kann. Lustige grafische Gestaltung sorgt dafur, daß das Uben Spass macht.
Preis: 25.- DM

GRUNDEIGENSCHAFTEN DES LICHTS (OPTIK): Diese 32KByte lange Programm erklart die Brechung, Reflexion, verschiedene Linsentypen... Ansprechende Grafik, Benutzereingaben, Menu und Ubungsaufgaben machen dieses Programm sehr lehrreich.
Preis: 39.- DM

NE555: Erklarung und Anwendung des Timer-ICs NE555 werden mit diesem Programm zum Kinderspiel. Tolle Grafik und ausfuhrliche Dokumentation heben dieses Programm hervor.
Preis: 25.- DM

PLOTTER: Plotter ist ein anspruchsvolles Mathemtikprogramm, das beliebige Funktionen zeichnet, Wertetabellen erstellt ...
Preis: 39.- DM

PHYSIK DES TRANSISTORS: Ahnlich aufwendig wie beim Programm 'Optik' werden Sie hier in die Grundeigenschaften des Transistors, verschiedene Schaltungen, Kennlinien u.s.w. eingewiesen.
Fur dieses Programm sind 32K Byte RAM erforderlich.
Preis: 39.- DM

LANDER-QUIZ: (Autor K. Weber) Ein ganz toll aufgemachtes Lernprogramm, das zugleich als Gesellschaftsspiel mit bis zu 8 Personen gespielt werden kann. Das Programm stellt jedem Spieler verschiedene Arten von Fragen, die teilweise durch eine Eingabe oder durch 'Zeigen' mit einem Kursor auf der Weltkarte beantwortet werden. Je nachdem, wie gut die Antwort ist, gibt es dann Punkte. (32K RAM erforderlich!)
Preis: 39.- DM

OPERATIONSVERSTARKER APPLIKATIONEN (OPAMAP, Autor: U. Gerber): Fur jeden, der sich mit Elektronik beschaftigt, ist dieses Programm eine wahre Freude ! Zwolf der wichtigsten Opamp-Schaltungen konnen mit diesem Programm komplett dimensioniert werden, wobei der Schaltplan graphisch dargestellt wird. Mitgeliefert wird ein 50-seitiges Lehrbuch, daß die Eigenschaften und Verwendung im allgemeine, und die Applikationsschaltungen im speziellen erklart. Sehr empfehlenswert !
(32K RAM erforderlich !)
Preis: 69.- DM

5.) DISKETTENPROGRAMME

Alle Programme, die Sie auf den vorangegangenen Seiten finden sind, bis auf 'Deathtrap' auf Diskette erhaltlich.
Selbstverstandlich brauchen Sie fur alle Programme ein Colour-Genie mit 32K RAM.
Bitte addieren Sie bei Bestellung auf Disk jeweils
DM 10.- zum Listenpreis fur die Diskette !

Im folgenden finden Sie eine Liste von Programmen, die speziell fur Diskette geschrieben sind:

TAPE-DISK-CONVERTER: Dieses Programm ermoglicht es Ihnen, Ihre Maschinensprache-Kassetten (sofern ungeschutzt) auf Diskette zu uberspielen, wobei eine Verschiebung im Speicher moglich ist. Außerdem kann man von Diskette auf Diskette, von Kassette auf Kassette und von Diskette auf Kassette uberspielen.
Preis: 45.- DM

COLZAP: Mit "Colzap" konnen Sie auf jeder (ungeschutzten) Colour-Genie-Diskette alle Sektoren lesen, beliebig andern (in ASCII oder Hex), Sektoren schreiben, den Disknamen anzeigen, den freien Platz auf der Diskette ausgeben lassen, u.a.m.
Preis: 45.- DM

ZEICHENEDITOR DISK: (Autor: J.Buchmuller) Dieses Programm ist rein in Maschinensprache geschrieben und ist daher sehr schnell und komfortabel. Es kann an mehreren Zeichensatzen gleichzeitig gearbeitet werden, die dann auch auf Diskette gesichert werden konnen.
Preis: 70.- DM

Ferner verfugbar (Funktionsbeschreibung siehe fruher):

COLOUR-ASSEMBLER Diskversion	Preis: 95.- DM
COLOUR-MONITOR III Diskversion	Preis: 105.- DM
COLOUR-FORTH Diskversion	Preis: 105.- DM
COLOUR-TEXT Diskverion	Preis: 105.- DM

6.) LITERATUR

Beachten Sie auch die fur das Colour-Genie lieferbaren Bucher:

Technisches-Handbuch: Enthalt die kompletten Schaltplane, Erklarungen, CRTC-Unterlagen, PSG-Unterlagen...
Preis: 49.- DM

DAS TCS Colour-Genie ROM LISTING (Autor K.Kampf): Unerlaßlich fur den Maschinensprache-Programmierer. In diesem Buch ist der komplette ROM und reservierte RAM aufgelistet und erklart. Insgesamt uber 300 Seiten ! 50 Seiten wertvolle Erklarungen ! Komplett und richtig erklarte Arithmetik-Routinen !
Preis: 59.- DM

DAS COLOUR-GENIE BUCH I: Auf uber 100 Seiten finden Sie hier interessante Programme - von Spielen bis zu Hilfsprogrammen, zum Teil auch in Maschinensprache - sowie wertvolle Tricks und Erklarungen. Ein Muß fur jeden Colour-Genie-Besitzer!
Preis: 39.- DM

DAS COLOUR-GENIE BUCH II: Noch umfangreicher als unser 'Bestseller' Das Colour-Genie Buch I' enthalt dieses Buch wieder viele interessante Tricks und Programme, darunter ein komplett gelistetes und erklartes Maschinensprache-Actionspiel !
Sehr interessant fur Diskettenbenutzer: Wie lese/schreibe ich Sektoren vom Basic aus, und wie kann ich meine Disketten schneller formatieren ?
Preis: 39.- DM

****
**** STAND DER TCS SOFTWARELISTE COLOUR-GENIE VOM 24.3.84 ****
****

BESTELLFORMULAR

Zur Bestellung von Colour-Genie-Programmen konnen Sie dieses Formular benutzten. Schicken Sie dieses an Ihren GENIE-Handler oder direkt an TCS Computer GmbH
Kolnstraße 4
5205 St. Augustin 2
Tel.: 02241 / 28071

Bankverbindung:
Raiffeisenbank Menden BLZ 37069507 Konto-Nr. 1525

Der Versand erfolgt per Nachnahme oder Vorauskasse auf obiges Konto (dies naturlich nicht, wenn Sie bei Ihrem Handler bestellen).

Ich bitte um Zusendung folgender TCS Colour-Genie-Programme:

Anzahl	Name des Programms	Preis/DM	Gesamt
		Gesamt:	DM

Meine Adresse ist:

Name: ....

Straße: ....

Ort: ....

Datum: .... Unterschrift: ....